



REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

**UNIVERSITE IBN KHALDOUN - TIARET**

# MEMOIRE

Présenté à :

FACULTÉ MATHÉMATIQUES ET INFORMATIQUE  
DÉPARTEMENT D'INFORMATIQUE

Pour l'obtention du diplôme de :

**MASTER**

Spécialité : Génie Logiciel

Par :

**KATI Yacine et NASRI Sami**

Sur le thème

---

## **Recommandation personnalisée par contexte**

---

Soutenu publiquement le 07 / 07 / 2019 à Tiaret devant le jury composé de :

Mr KOUADRIA Abderrahmane

Université Ibn Khaldoun

Président

Mr KHAROUBI Sahraoui

Université Ibn Khaldoun

Encadreur

Mr BOUDAA Boudjema

Université Ibn Khaldoun

Examineur

## REMERCIEMENTS

*Nous tenons à remercier ALLAH, le tout puissant de nous avoir  
donné courage et volonté pour accomplir ce modeste travail.*

*Nous tenons à exprimer tous nos reconnaissances à tous ceux qui  
ont contribué de près ou de loin à la réalisation de ce travail.*

*Nous tenons à présenter notre profonde gratitude à notre encadreur  
Mr KHAROUBI Sahraoui pour sa disponibilité, ses remarques et  
conseils.*

*On voudrait remercier tous les responsables des entreprises de  
l'échantillon pour leurs contributions.*

*Nos remerciements vont également à l'ensemble de nos enseignants  
de la spécialité MI qui nous ont apporté soutiens et conseils.*

*Et à tous ceux qui ont participé de près ou de loin à la réalisation de  
ce travail.*

# Table des matières

<b>REMERCIEMENTS .....</b>	<b>I</b>
<b>Résumé .....</b>	<b>VII</b>
<b>Abstract.....</b>	<b>VII</b>
<b>Introduction Générale .....</b>	<b>1</b>
<b>1. Introduction.....</b>	<b>4</b>
<b>2. Problématique de surcharge d’information .....</b>	<b>4</b>
<b>3. Définition du filtrage d’information .....</b>	<b>5</b>
<b>5. Fonctionnement d'un système de filtrage .....</b>	<b>6</b>
<b>6. Types d'un système de filtrage d'information.....</b>	<b>7</b>
<b>6.1 Filtrage Cognitif.....</b>	<b>7</b>
<b>6.1.1 Définition.....</b>	<b>7</b>
<b>6.1.2 Descripteur d’article et profil utilisateur .....</b>	<b>8</b>
<b>6.1.2.1 Descripteur d’article.....</b>	<b>9</b>
<b>6.1.2.2 Descripteur de profil d'utilisateur.....</b>	<b>9</b>
<b>6.1.3 Limites du filtrage cognitif .....</b>	<b>9</b>
<b>6.2 Filtrage collaboratif.....</b>	<b>10</b>
<b>6.2.1 Définition.....</b>	<b>10</b>
<b>6.2.2 Architecture générale.....</b>	<b>11</b>
<b>6.2.3 Avantages du filtrage collaboratif .....</b>	<b>12</b>
<b>6.2.4 Evaluation des recommandations.....</b>	<b>12</b>
<b>6.2.5 Apports par rapport au filtrage cognitif.....</b>	<b>13</b>
<b>6.2.6 Limites du filtrage collaboratif .....</b>	<b>14</b>
<b>6.3 Filtrage Collaboratif vs Filtrage Cognitif .....</b>	<b>15</b>
<b>6.4 Filtrage hybride .....</b>	<b>16</b>
<b>6.5 Calcul de prédiction .....</b>	<b>18</b>
<b>6.5.1 Prédiction dans le filtrage cognitif.....</b>	<b>18</b>
<b>6.5.2 Prédiction dans le filtrage collaboratif.....</b>	<b>18</b>
<b>6.6 Autres formes de filtrage .....</b>	<b>21</b>
<b>6.6.1 Filtrage actif.....</b>	<b>21</b>
<b>6.6.2 Filtrage différé .....</b>	<b>21</b>

6.6.3 Filtrage adaptatif.....	21
6.7 Systèmes de filtrages renommés.....	22
6.7.1 Système COCoFil .....	22
6.7.2 Système de Maltz et Elrich .....	23
6.7.3 Système Tapestry .....	23
6.7.4 Système ReferralWeb .....	23
Conclusion .....	24
1. Introduction.....	26
2. Profil Utilisateur.....	26
2.1 Définition .....	26
2.2 Profils à court-terme et à long-terme.....	27
2.3 Modèles de Représentation du profil utilisateur .....	27
2.3.1 Modèle vectoriel .....	27
2.3.2 Modèle sémantique à base d'ontologie .....	28
2.3.3 Modèle Multidimensionnel.....	29
2.3.4 Modèle matrice utilisateurs-items .....	29
2.4 Acquisition d'Information .....	29
2.4.1 Acquisition explicite .....	30
2.4.2 Acquisition implicite .....	30
2.4.3 Approche hybride.....	30
2.5 Mise à jour de profils utilisateurs .....	31
3. Définition de Contexte .....	32
4. Recommandation par Contexte .....	32
5. Dimensions du contexte .....	33
5.1 Contexte lié à l'utilisateur .....	33
5.2 Contexte lié à la requête.....	33
6. Emergence du contexte.....	34
6.1 Pré-filtrage contextuel.....	34
6.2 Post-filtrage contextuel .....	35
6.3 Modélisation directe du contexte .....	35
7. Proposition de recommandation hybride à base de prédiction des liens.....	36
7.1 Les Graphes.....	36
7.1.1 Définition.....	36

7.1.2 Types de graphes .....	36
7.2 Prédiction des liens dans les graphes bipartis.....	38
8. Conclusion .....	41
1. Environnement de travail.....	44
1.1 Java .....	44
1.2 Netbeans .....	45
2. Mesures de tests.....	45
3. Jeux de Données .....	50
3.1 MovieLens .....	50
4. Résultats et discussion .....	52
4.1 Déroulement de l'exécution.....	52
Bibliographie .....	58

## Liste des figures

Figure 1. Modèle général pour le filtrage d'information .....	6
Figure 2. Modèles de recherche et de filtrage d'information.....	7
Figure 3. Filtrage basé sur le contenu .....	8
Figure 4. Principe général du filtrage collaboratif .....	11
Figure 5. Filtrage collaboratif.....	11
Figure 6. Hybridation classique de filtrage .....	18
Figure 7. Exemple du profil utilisateur représenté par le modèle d'ontologie....	28
Figure 8. Modélisation Contextuelle.....	32
Figure 9. Pré-Filtrage Contextuel.....	34
Figure 10. Post-Filtrage Contextuel .....	35
Figure 11. Modélisation Contextuelle.....	36
Figure 12. Un graphe orienté avec trois sommets et quatre arêtes. ....	37
Figure 13. Un graphe pondéré avec dix sommets et douze arêtes. ....	38
Figure 14. Projection de graphe biparti user-item.....	41
Figure 15. Graphe de ROC.....	49
Figure 16. Fenêtre principale .....	52
Figure 17. Fenêtre d'accès .....	53
Figure 18. Fenêtre filtrage collaboratif .....	53
Figure 19. Fenêtre prediction des liens .....	54
Figure 20. Fenêtre comparaison graphique.....	55
Figure 21. Fenêtre résultats des prédictions .....	55
Figure 22. Résultats du MAE (Matrice 10x10).....	56
Figure 23. Résultats du MAE (Matrice 20x20).....	56

## Liste des tableaux

Tableau 1. Filtrage collaboratif basé sur les notes .....	13
Tableau 2. Similarité dans un graphe biparti .....	40
Tableau 3. Classes des items .....	46

# Résumé

Actuellement le filtrage Collaboratif a été répandu usagé pour résoudre le problème de surcharge des informations. Cependant ici reste encore une limitation majeure, la matrice creuse « sparsity ». Dans ce travail, nous nous intéressons à la prédiction des liens dans les graphes bipartis. La technique de prédiction réduit le problème de matrice creuse et améliore la précision des prédictions. Les différentes stratégies de calcul de similarité et de prédiction des évaluations peuvent être basées sur les utilisateurs et les items. Parmi les techniques de prédiction des liens nous avons utilisé les formules du coefficient de Jaccard et Corrélation de Pearson qui sont jugées intéressantes et efficaces.

**Mots clés :** Filtrage Collaboratif, Corrélation de Pearson , Jaccard, Recommandation

# Abstract

Currently Collaborative filtering has been widely used to solve the problem of information overload. However, there is still a major limitation, the matrix "sparsity". In this work, we are interested in the link prediction in bipartite graphs. The prediction technique reduces the the sparsity problem in the matrix and improves the accuracy of predictions. Different strategies for calculating similarity and predicting evaluations can be based on users and items. Among the link prediction techniques, we used the Jaccard's coefficient and Pearson correlation formulas, which are considered interesting and effective.

**Keywords :** Collaborative Filtering, Jaccard, Pearson Correlation, Recommendation

---

# **Introduction Générale**

---

## **Introduction Générale**

Le volume d'information accessible via le Web est en pleine croissance, cependant toute information qui s'y trouve n'est pas nécessairement utile ou pertinente. Donc les modalités d'accès à cette information évoluent aussi pour répondre aux nouveaux besoins.

A la différence des moteurs de recherche où l'accès est actif, les systèmes de filtrage présentent un accès passif à l'information. Un système de filtrage d'information permet à partir d'un flot d'informations entrant d'extraire et de présenter les seuls documents susceptibles d'intéresser un utilisateur ou un groupe d'utilisateurs ayant des besoins en information relativement stables appelés profil.

Selon la façon dont l'information est obtenue, nous allons établir une distinction entre les systèmes :

**Explicite** : L'utilisateur note les items soit, en donnant une valeur numérique (sur une échelle de 1 à 5 par exemple), ou bien une valeur qualitative (excellent, bon, moyen, mauvais).

**Implicites** : le système induit la note donnée à un item à travers les actions effectués par l'utilisateur.

Les deux grandes approches des systèmes de filtrage sont :

**Le filtrage « basé sur le contenu »** : le profil d'utilisateur est constitué de ses centres d'intérêts exprimés à travers d'une liste de thèmes .Le système réalise une interrogation permanente du flot de documents entrants avec le profil de l'utilisateur en guise de requête. **Le filtrage « collaboratif »** : cette approche est perçue comme un processus de recherche et d'exploitation de corrélation entre les utilisateurs d'un système. Pour calculer ces corrélations, il est donc nécessaire de construire un profil utilisateur.

Le profil de l'utilisateur ici est un ensemble d'évaluations effectuées par l'utilisateur sur différents items.

Le système garde une table contenant les évaluations des utilisateurs, appelé une *matrice d'évaluation*. Cette matrice est traitée pour obtenir l'évaluation de nouveaux éléments, et ainsi recommander des nouveaux éléments pour les utilisateurs.

La façon de traitement des données nous permettant de différencier entre deux types d'algorithmes de filtrage collaboratif :

- ✓ *Les algorithmes basés mémoire* traite toute la base de données des évaluations pour trouver une ressemblance entre les utilisateurs,
- ✓ *Le filtrage collaboratif basé modèle* permet de créer des modèles souvent par apprentissage supervisé de manière à l'instancier afin de calculer une prédiction pour un utilisateur donné.

Dans notre travail, nous nous sommes penchés sur le filtrage Collaboratif basé mémoire en se basant sur la prédiction des liens dans les graphes bipartis visant à résoudre le problème de la sparsité.

Ce travail est organisé en deux parties. La première partie étale l'état de l'art Afin de bien comprendre les différentes évolutions techniques dans ce domaine, dans son premier chapitre du domaine du filtrage d'information, quant au deuxième chapitre, il présente la Prédiction des liens qui est le crucial dans notre travail.

La deuxième partie est réservée à l'Expérimentation et l'implémentation Dans le chapitre III, nous présentons notre proposition et ses résultats que nous évaluons en utilisant des techniques de mesure. Nous terminons par une conclusion et la description des perspectives.

---

# **Chapitre 1 : Filtrage d'information (Généralités)**

---

## **1. Introduction**

Avec l'explosion de la masse d'information sur l'Internet, il est devenu plus que jamais indispensable d'organiser ce gros volume d'information presque illimité afin d'en tirer le meilleur profit. Les recherches actuelles visent à concevoir des mécanismes capables de faire parvenir continuellement à l'utilisateur l'information qui l'intéresse au lieu que celui-ci dernier dépense son temps à la chercher, c'est la fonction du filtrage d'information [BOU05] .

Le filtrage de l'information est une activité, que nous avons toujours exercé dans notre vie quotidienne, lorsque nous sélectionnons un article ou un texte du journal selon nos centres intérêts ou bien lorsque nous achetons seulement qu'une certaine catégorie de magazines ou en naviguant sur certains sites ou en ouvrant quelques types de messages, nous faisons du filtrage. Le filtrage est un processus naturel car l'être humain de tout temps et de par sa nature a tendance à ne pas prendre et accepter toujours tous ce qui lui est soumis, mais cherche plutôt ce qui convient et répond à son besoin .

Avec l'expansion de Web et l'avènement du document électronique le filtrage de l'information est devenu impossible avec les seuls efforts de l'être humain, vu ses limites dans la maîtrise de cette masse presque illimitée d'information et du grand nombre de document qui circule sur la toile. D'où l'idée de déléguer la fonction de filtrer automatiquement l'information par des systèmes mis en place à cet effet.

Dans ce chapitre, nous présentons les diverses méthodes de filtrage d'information proposées dans la littérature. En particulier, nous décrivons les deux principales approches qui sont le filtrage collaboratif et le filtrage par contenu.

## **2. Problématique de surcharge d'information**

Le cadre de ce travail concerne les sites web de contenu multimédia, plus particulièrement les sites de visionnage de vidéos et films en ligne. Ces domaines sont particulièrement touchés par le problème de la surcharge d'information car ils renferment un volume d'information assez conséquent. A titre d'exemple, le site de MovieLens comprend près de 27000 différents films à disposition du grand public. De ce fait, les utilisateurs sont confrontés à plusieurs problèmes : ils sont submergés par le nombre très important de choix possibles dans l'espace qu'ils explorent. L'exploitation de cette longue liste d'options est très complexe pour les utilisateurs, qui doivent

passer beaucoup de temps pour sélectionner les films qui correspondent le plus à leurs intérêts. De plus, les utilisateurs ne savent pas forcément ce qu'ils devraient voir ou ce qu'ils pourraient apprécier, le parcours qu'ils font n'est alors en général pas réfléchi, ou bien ils se limitent à voir les items les plus populaires. En conséquence, ils peuvent perdre du temps en regardant des films ou des vidéos qui ne les intéressent pas. Inversement, ils peuvent manquer des films ou des vidéos qui auraient pu les intéresser.

Un des domaines de recherche principaux relatifs à la problématique de la surcharge d'information est le domaine de la recherche d'information. Le principe général est d'élaborer des méthodes et des algorithmes afin de rechercher des ressources (par exemple, des pages web, des articles, des livres et du contenu multimédia) en fonction de requêtes formulées par des utilisateurs. Il n'est cependant pas toujours évident pour un utilisateur de savoir comment exprimer sa demande. De plus, sa requête correspond généralement à une quantité importante de ressources et il est difficile de savoir quels résultats lui présenter en premier.

Un autre domaine de recherche relatif à cette problématique est le domaine des systèmes de recommandation. Ces systèmes sont capables de fournir des recommandations adaptées aux préférences et aux besoins des utilisateurs. Initialement conçus pour la recommandation de ressources web, livres, articles, etc. les systèmes de recommandation sont devenus de plus en plus populaires et sont aujourd'hui un composant principal de beaucoup d'applications dans différents domaines. Un avantage très conséquent des systèmes de recommandation est que l'utilisateur n'a pas besoin de formuler de requêtes. Sa seule requête est implicite. Les systèmes de recommandation peuvent être classés en deux types d'approches : les approches basées sur le contenu et les approches basées sur le filtrage collaboratif. Les recommandations basées sur le contenu sont effectuées en identifiant les ressources similaires à celles appréciées par un utilisateur en fonction de leur contenu. Les approches basées sur le filtrage collaboratif, quant à elles, permettent de fournir des recommandations à un utilisateur sans forcément considérer le contenu des ressources, mais en se basant sur l'analyse du comportement et/ou des appréciations de l'utilisateur afin de recommander les ressources qui ont été appréciées par d'autres utilisateurs ayant des goûts similaires.

### **3. Définition du filtrage d'information**

Le filtrage d'information est l'expression utilisée pour décrire une variété de processus ayant pour but de fournir des informations à des personnes, informations en adéquation avec leurs centres d'intérêt [GAL05]. Ces intérêts sont préalablement décrit en besoins d'informations, appelés profils.

Les systèmes de filtrage d'information fournissent à leurs utilisateurs un flot continu de documents, sans qu'ils aient à exprimer explicitement ce qu'ils cherchent, en contrepartie, il faut que le système connaisse le profil de l'utilisateur à travers ses centres d'intérêt [GAL05].

Les domaines d'application du filtrage de l'information sont divers et d'une importance capitale sur le plan économique, citons parmi eux : mailing list, Usenet News, filtrage des e-mails, filtrage dans le Web.

Le filtrage des informations sur le Web est réalisé par la détermination des contenus à envoyer au visiteur d'un site Internet en fonction des critères spécifiques qui lui sont propres et qui constituent son profil. Le profil peut être statique et déterminé a priori lors de la première consultation, en remplissant un formulaire par exemple ou dynamique, conçu et maintenu au fur et à mesure des consultations par l'observation du comportement de l'internaute lors de ses interactions avec le système.

#### 4. Architecture d'un système de filtrage d'information

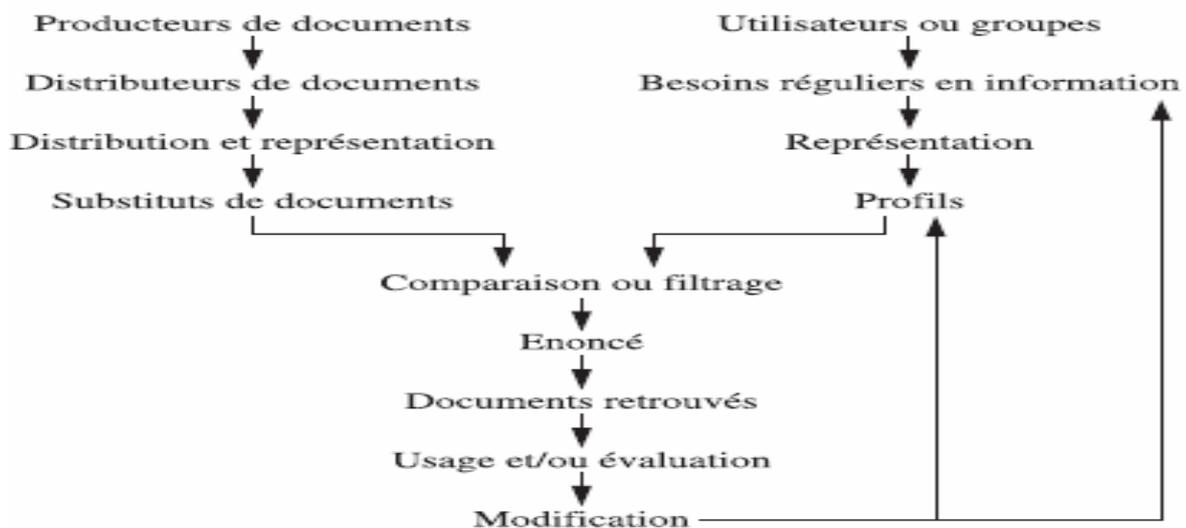


Figure 1. Modèle général pour le filtrage d'information

#### 5. Fonctionnement d'un système de filtrage

La recherche et le filtrage d'information, consiste à modéliser, concevoir et mettre en application des systèmes capables de traiter un flux d'informations au fur et à mesure de leurs arrivées pour n'en sélectionner et diffuser que celles qui sont pertinentes. La recherche d'information inclue trois fonctions principales qui sont l'organisation et l'indexation de la requête de l'utilisateur et des documents qui sont des collections

relativement stables, ainsi que la méthode utilisée pour faire la comparaison entre les deux.

En contrepartie le filtrage d'information traite des documents entrants qui sont représentés sous un format spécifique et des profils qui sont des sortes de requêtes à long terme, ainsi que la méthode employée pour faire l'appariement entre les documents et les profils. Cet appariement s'inspire des techniques classiquement utilisées en recherche d'information.

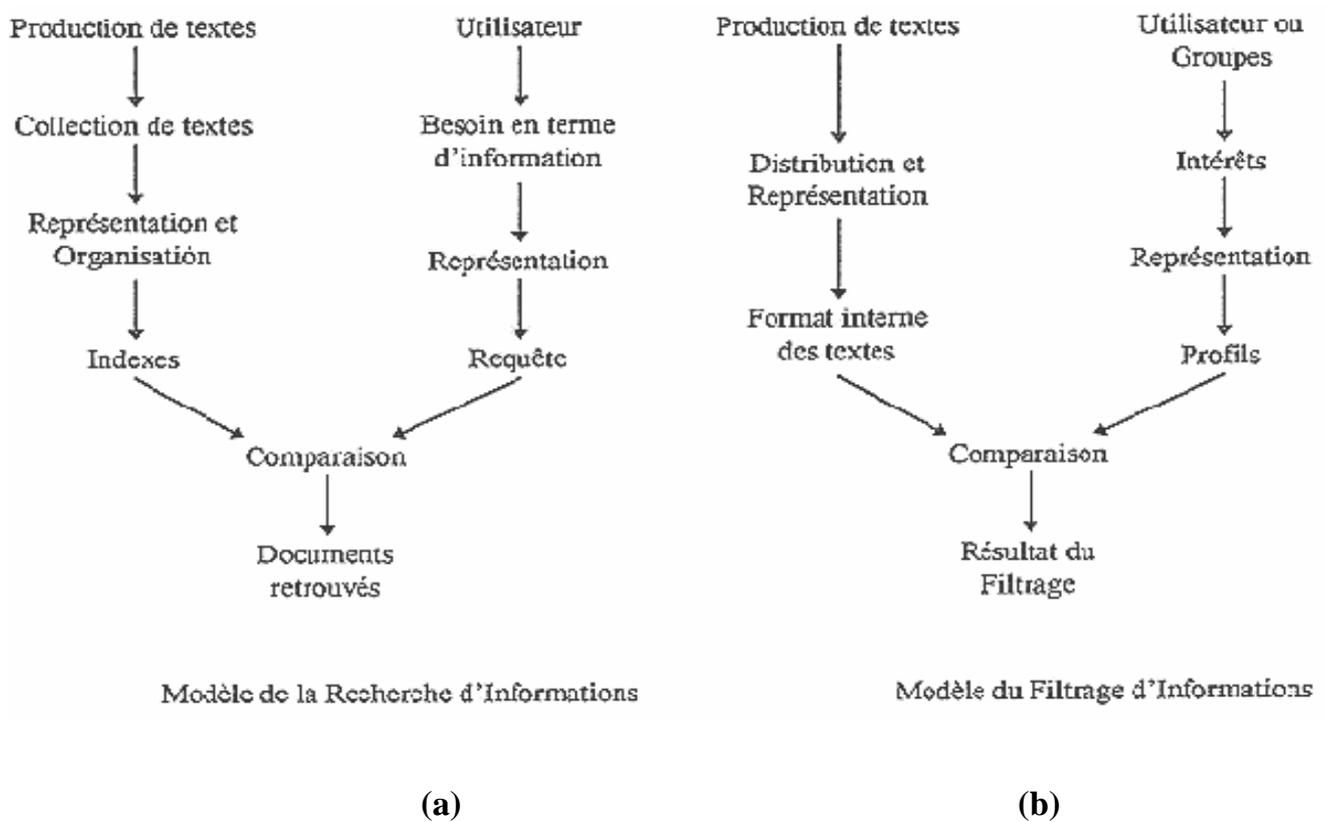


Figure 2. Modèles de recherche et de filtrage d'information

## 6. Types d'un système de filtrage d'information

### 6.1 Filtrage Cognitif

#### 6.1.1 Définition

Dans le filtrage cognitif, la sélection des documents est basée sur l'analyse de

leur contenu, le document n'est alors proposé à l'utilisateur que s'il répond aux mêmes thèmes intéressant celui-ci. Le filtrage par le contenu peut être perçu comme un système de recherche d'informations qui se sert de la requête pour extraire les informations pertinentes du corpus interrogé, parallèlement le filtrage par le contenu, emploie plutôt un profil utilisateur qui est une sorte de requête dynamique à long terme pour filtrer en permanence un flot évolutif de documents entrants.

Les systèmes de filtrage d'information basée sur le contenu (ou cognitifs) des documents est l'approche la plus anciennement utilisée dans le domaine du filtrage, elle trouve ses racines dans le monde de la recherche d'information et utilise beaucoup de ses principes. Les documents entrants sont indexés par des termes et comparés au profil utilisateur représenté sous la forme d'un ensemble de thèmes décrivant ses centres d'intérêt.

Dans ce genre de système, chaque utilisateur opère indépendamment des autres comme s'il était en train de faire une recherche d'information classique.

Les fonctionnalités principales du filtrage par le contenu visent la sélection des documents pertinents par rapport au profil et à mettre à jour ce dernier en fonction du retour de pertinence émis par l'utilisateur sur les documents qu'il a reçu. La mise à jour se fera par l'insertion des nouveaux thèmes abordés par les documents jugés pertinents.

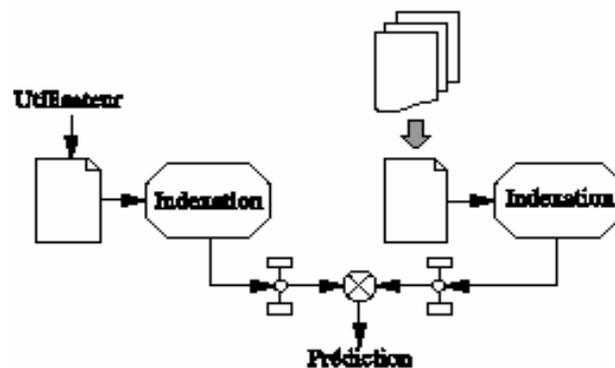


Figure 3. Filtrage basé sur le contenu

### 6.1.2 Descripteur d'article et profil utilisateur

L'algorithme de filtrage basé sur le contenu peut réaliser le matching entre un descripteur de contenu (comme par exemple, documents, News, Url, livre, etc.) et un profile utilisateur et détermine le degré de pertinence de chaque article (ou contenu) pour les utilisateurs potentiels. Si de nombreux articles s'accumulent dans un certain laps de temps, l'algorithme de filtrage de contenu peut ordonner les articles en fonction de leur pertinence pour chacun des utilisateurs potentiels.

### 6.1.2.1 Descripteur d'article

Un descripteur d'article se compose d'un ensemble de concepts qui peuvent être représentés par une ontologie de domaine. Les concepts qui représentent un élément sont les plus spécialisés dans une branche de la hiérarchie. Par exemple, si un article traite le « sport » et en particulier le « football », il est représenté par «football» seulement.

De toute évidence, un article peut être représenté avec de nombreux concepts de l'ontologie, chaque concept peut apparaître dans n'importe quelle branche de la hiérarchie de l'ontologie et à tout niveau – cela dépend du contenu réel de cet article. Il est à noter que le profil peut inclure des concepts frères, c'est-à-dire les fils d'un même concept. Par exemple, un profil d'article peut inclure à la fois « football » et « basketball » les fils de « sport ».

### 6.1.2.2 Descripteur de profil d'utilisateur

Un profil d'utilisateur basé sur le contenu se compose d'une liste représentant ses préférences (ses intérêts).

le profil utilisateur est défini par l'ensemble des préférences contextuelles implicites (déduts automatiquement par le système en analysant les logs d'interactions de l'utilisateur). Cette modélisation permet de capturer le fait qu'un utilisateur puisse avoir différents scores de préférence sur le même objet en fonction de son contexte courant [SOL12].

Ensuite, le profil de l'utilisateur est mis à jour au fil du le temps en fonction de la réaction implicite de l'utilisateur : lorsque l'utilisateur lit un article et le trouve intéressant, les concepts dans le descripteur de cet article qui n'étaient pas initialement dans le profil utilisateur, sont ajoutés à celui-ci, et les poids de tous les concepts dans ce profil sont recalculés en fonction du nombre de clicks [SOL12].

### 6.1.3 Limites du filtrage cognitif

- Difficulté d'indexation de documents multimédias du fait que le filtrage par le contenu se base sur un profil qui exprime le besoin de l'utilisateur sur le plan thématique, exactement comme une requête utilisée dans le cadre de la recherche d'information.

Quel que soit la diversité du profil de l'utilisateur, il est toujours exclusivement décrit que par des termes qui seront comparés aux termes qui indexent les documents, or les documents multimédias (image, son, vidéo) sont difficiles à indexer, ce qui constitue une grande restriction et un véritable obstacle, pour l'application de cette technique.

- Le filtrage par le contenu ne permet pas l'intégration d'autres critères de pertinence autre que le facteur thématique, or il existe d'autres facteurs de pertinence tels que l'adéquation entre le public visé par l'auteur et l'utilisateur ou encore la qualité scientifique des faits présentés, la fiabilité de la source d'information, le degré de précision des faits présentés.
- L'effet de l'entonnoir, en effet le besoin de l'utilisateur en informations est de plus en plus spécifique or les systèmes actuels ne permettent pas dans certains cas de filtrer tous les documents surtout ceux qui sont proches thématiquement mais décrit d'une manière différente [MAL95].

## 6.2 Filtrage collaboratif

### 6.2.1 Définition

Le filtrage collaboratif vient renforcer l'idée que les personnes à la recherche d'informations devraient pouvoir se servir de ce que d'autres ont déjà trouvé et évalué. Dans la plupart des communautés partageant les mêmes centres d'intérêt, les personnes se recommandent ou s'échangent régulièrement des documents parmi leurs amis ou collègues. Le filtrage collaboratif consiste à filtrer les documents ou toute autre ressource du flux entrant en se basant sur l'opinion que, chaque utilisateur de la communauté a porté dessus. Tout document qu'il aura alors jugé intéressant, sera diffusé à l'ensemble des utilisateurs ayant eu des opinions similaires par le passé [GAL05].

Dans la figure 4, nous schématisons le principe du filtrage collaboratif. Le document  $d$  sera recommandé à l'utilisateur  $u$ , car ce document est apprécié par la communauté  $G$  où se trouve l'utilisateur.

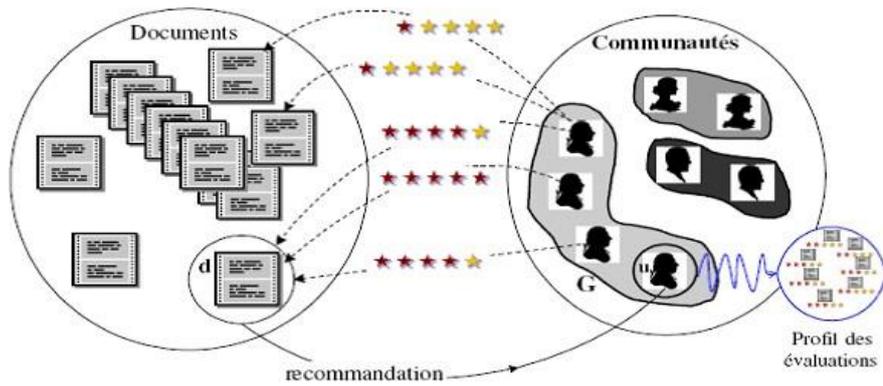


Figure 4. Principe général du filtrage collaboratif

### 6.2.2 Architecture générale

Un système de filtrage collaboratif est constitué de trois principales fonctionnalités : le calcul de la prédiction de l'évaluation du document par l'utilisateur, le calcul de la proximité entre les utilisateurs en plus de la mise à jour périodique des profils utilisateurs à chaque nouvelle évaluation de documents.

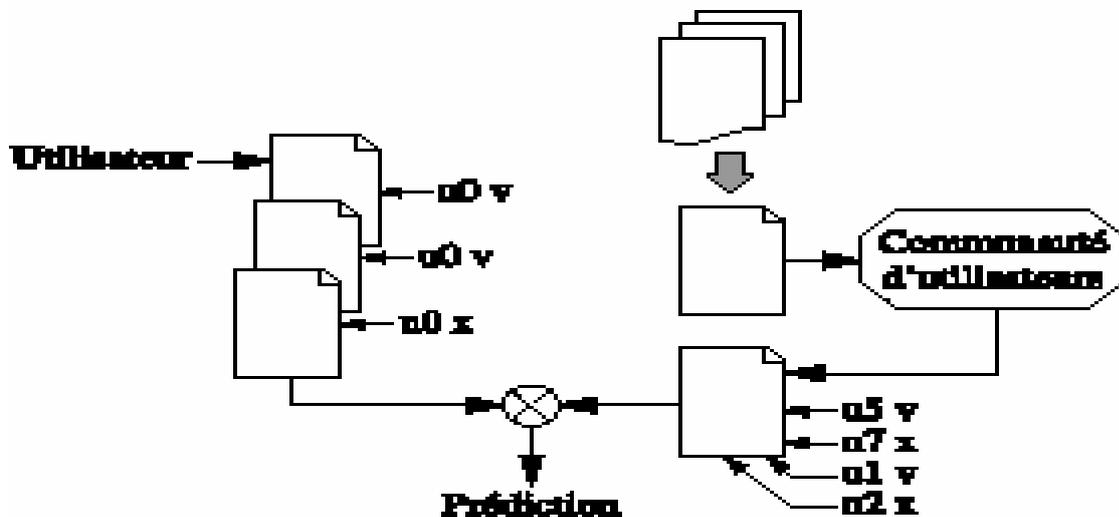


Figure 5. Filtrage collaboratif

Certains systèmes de filtrage collaboratifs offre d'autres types d'interaction avec l'utilisateur à travers ce qu'on appelle le « filtrage actif » dans lequel l'utilisateur prend l'initiative de proposer des documents à d'autres membres de la communauté.

Cette approche est particulièrement intéressante pour le démarrage du système, pour faire augmenter les chances de regroupement des profils par le calcul des prédictions. En effet L'utilisateur qui reçoit un document d'un autre, il est invité à son tour à prononcer sa propre évaluation, ce qui fait que son profil et celui de l'émetteur peuvent se regrouper.

### 6.2.3 Avantages du filtrage collaboratif

Le filtrage collaboratif apporte des solutions aux insuffisances du filtrage par le contenu avec la contribution et la collaboration d'une communauté d'utilisateurs utilisant le système en permanence et partageant les mêmes centres d'intérêts à long terme.

Le filtrage collaboratif n'analyse pas le contenu du document qui n'est d'ailleurs connu que par son identifiant, mais utilise plutôt les évaluations des utilisateurs du système. Ces évaluations sont déployées par des méthodes statistiques pour prévenir ce que pourraient être les intérêts des utilisateurs et les proposer par la suite à d'autres sur la base de la proximité de leurs profils et du partage des mêmes goûts et intérêts [BER03].

Les systèmes de filtrage d'informations collaboratifs donnent sans doute des résultats plus pertinents et plus personnalisés du fait de l'implication de l'utilisateur dans le processus d'évaluation des documents. L'évaluation d'un document est effectuée en générale par l'attribution d'une note proposée par l'utilisateur selon son appréciation de son contenu.

Le jugement de pertinence de l'utilisateur est plus crédible que celui qu'aurait porté un système de traitement automatique, car les ordinateurs sont confortés à de nombreuses difficultés que l'être humain peut surmonter facilement tel que la compréhension du texte, la synonymie, la polysémie. Le document évalué ainsi que la note qui lui aura été attribué, feront partie par la suite du nouveau profil de l'utilisateur [BOU05].

### 6.2.4 Evaluation des recommandations

Selon le principe de base du filtrage collaboratif, les utilisateurs doivent fournir leurs évaluations sur des documents afin que le système forme les communautés.

Evaluer une recommandation peut se faire de façon explicite ou implicite, comme suit [SOL12] :

- **Explicite** : L'utilisateur donne une valeur numérique sur une échelle donnée (par exemple, de 1 à 5, ou de 1 à 10, etc.), ou bien, une valeur qualitative de satisfaction, par exemple, *mauvaise, moyenne, bonne et excellente*.
- **Implicite** : Le système induit la satisfaction de l'utilisateur à travers ses actions. Par exemple, le système estimera qu'une recommandation supprimée correspond à une évaluation très mauvaise, alors qu'une recommandation imprimée ou sauvegardée peut être interprétée comme une bonne évaluation. Il faut par ailleurs noter que les recommandations qu'évalue un utilisateur peuvent être générées par le système et/ou choisies par l'utilisateur lui-même.

**Tableau 1. Filtrage collaboratif basé sur les notes**

	Article 1	Article 2	Article 3	Article 4	Article 5 considéré
Utilisateur 1	-	-	7	6	???
Utilisateur 2	-	-	5	6	1
Utilisateur 3	-	-	6	6	3
Utilisateur 4	7	5	-	-	6
Utilisateur 5	7	6	-	-	4

Ici, on cherche la note que l'utilisateur 1 va attribuer à l'article 5, en fonction des notes que les autres utilisateurs ont attribué à l'article, et de la similarité entre leurs vecteurs de notes sur les autres articles.

### 6.2.5 Apports par rapport au filtrage cognitif

Le filtrage collaboratif permet de contourner les limitations du filtrage cognitif :

- Le problème de l'indexation des documents multimédia ne se pose plus, du fait que la sélection et le filtrage ne s'opèrent plus par le contenu mais plutôt en fonction des opinions des utilisateurs sur les documents.
- Les différentes opinions des utilisateurs sur les documents sont un excellent moyen pour mieux les indexer afin de pouvoir les extraire convenablement ultérieurement. De plus cette indexation parallèle émerge d'autres facteurs de pertinence autres que le contenu thématique du document, tel que sa qualité intellectuelle ou informationnelle, l'adéquation de son contenu avec ce que recherche réellement

l'utilisateur, la précision des faits qui y sont évoqués ou encore la fiabilité de la source qui les, a rapporté.

- Les documents ne sont plus filtrés par leur contenu, mais plutôt sur la base d'opinions d'utilisateurs de la communauté, quels que soient les termes qui indexent leur contenu, ils ne constituent plus un obstacle pour qu'un utilisateur puisse les recevoir, car il suffit seulement qu'ils aient été jugés auparavant pertinents par la communauté. De ce fait le problème de l'entonnoir est ainsi radié et dès lors l'utilisateur pourra à son tour modifier son profil en y ajoutant de nouveaux termes et transmettra un retour de pertinence positif sur les documents qu'il a reçu [BER03].

Le filtrage collaboratif remédie aux limitations du filtrage par le contenu mais en revanche, il présente quelques insuffisances, d'où l'idée d'adopter des approches hybrides qui compilent les apports des deux catégories. La tendance actuelle de la recherche vise donc à tirer profit des avantages du filtrage collaboratif et du filtrage basé sur le contenu, tout en évitant leurs inconvénients inhérents [BEN17].

### 6.2.6 Limites du filtrage collaboratif

- Le système dépend fortement de son utilisation et du comportement de l'ensemble des utilisateurs. Un utilisateur qui ne contribue pas à l'évaluation des documents reçus voit la qualité des réponses et la pertinence des recommandations, baisser non seulement pour lui, mais le fait également baisser pour les autres utilisateurs de son voisinage.
- L'utilisateur évalue l'intérêt du système par rapport à ce qu'il pourrait en tirer en effet lorsqu'il ressent un rapport coût/bénéfice déficitaire ou perçoit une constante baisse dans la pertinence, des documents et des recommandations, il peut se décourager, ce qui le conduit inévitablement à l'abandon du système.
- Le démarrage à froid est un grand problème pour le filtrage collaboratif, le système doit avoir un minimum d'évaluations des utilisateurs, pour que le calcul de prédiction prenne tout son importance pour filtrer efficacement des documents pour un utilisateur, or les profils au départ sont vides et une période d'apprentissage plus au moins longue est toutefois nécessaire pour les constituer, période durant laquelle beaucoup d'utilisateurs peuvent abandonner le système.

- Les utilisateurs qui ont des centres d'intérêts singuliers, risquent de ne pas recevoir des propositions et des recommandations par le système et par conséquent ils se retrouveront inévitablement isolés des autres utilisateurs du système.
- Le problème des nouveaux documents qui ne peuvent être diffusés aux utilisateurs concernés du fait qu'ils n'ont pas encore été évalués.

### 6.3 Filtrage Collaboratif vs Filtrage Cognitif

À la base, nous avons des informations sur le parcours hypertextuel de nos clients. Dans un premier temps, nous en avons déduit quelles sont les pages qui ont plu, quelles sont celles qui ont déplu, et quelles sont celles qui n'ont pas été visitées. Ceci nous a permis de former un premier profil pour nos clients [SOL12]. Pour cela, à chaque utilisateur nous associons à un vecteur de notes sur les pages qu'il a parcourues. C'est cette représentation de profil qui est utilisée pour le filtrage collaboratif. Par contre, la méthode d'analyse de contenu ne s'arrête pas là. Pour chaque page identifiée, une analyse en profondeur du texte va permettre d'en extraire les caractéristiques essentielles, pour tenter de mieux comprendre ce qui plait ou déplaît dans la page au client, et former un autre profil en associant en plus à chaque utilisateur un vecteur des thèmes (et éventuellement des mots) les plus fréquemment rencontrés dans ses lectures, et constituer ainsi une mémoire des centres d'intérêts du client. Chacune des approches possède ses limitations. Celles de l'analyse basée sur le contenu sont les suivantes :

- Avant d'analyser le contenu, les articles doivent, dans un premier temps, être mis sous une forme convenable pour l'algorithme. Cependant, des nouvelles technologies telles le son, les photos, ou la vidéo, etc. ne peuvent être analysées automatiquement pour obtenir des informations suffisamment pertinentes.
- Etant donné que relativement peu de textes ont des vecteurs vraiment similaires, le système recommande souvent ce que l'utilisateur a déjà vu auparavant.
- À part si le système utilise un niveau d'analyse linguistique très poussé, il ne peut filtrer des articles en se basant sur leur qualité, style ou point de vue. Par exemple, si on se restreint à une approche bag-of-words, et si deux articles contiennent les mêmes termes, on ne pourra distinguer entre un bien écrit et un autre mal écrit.

Au contraire, le filtrage collaboratif n'utilisant aucune information concernant le contenu des articles (texte, image, musique, etc.), les articles filtrés n'ont pas besoin d'être préalablement analysés grammaticalement [SOL12]. De plus, ce système étant basé sur l'évaluation des objets par les utilisateurs, les articles recommandés peuvent alors être très différents de ceux que l'utilisateur a déjà vus auparavant, et les recommandations sont basées sur une évaluation implicite de la qualité des articles, perçue par les utilisateurs. Mais par contre, trois limites sont présentes pour cette approche :

- La première est que la recommandation à un client dépend des autres clients.
- Le second est qu'il faut associer à chaque client la liste des textes lus.
- Et la troisième est le fait que le système repose sur l'hypothèse implicite que toutes les caractéristiques d'un document sont de la même importance pour l'évaluation de ce document par un utilisateur. Or, deux utilisateurs peuvent donner tous deux une bonne (ou inversement une mauvaise) note à un même document pour des raisons complètement différentes.

### 6.4 Filtrage hybride

Chacune des approches de filtrage présentées présente des avantages ainsi que des inconvénients. Par exemple, les systèmes basés sur le contenu ont besoin d'un riche historique d'interactions pour pouvoir fonctionner ; le système ne pourra pas fournir des recommandations de qualité pour un utilisateur fraîchement inscrit. D'un autre côté, les systèmes du type filtrage collaboratif ont besoin de l'existence d'une large base d'interactions sur l'ensemble du catalogue d'objets/produits du système afin de pouvoir calculer des rapprochements entre les utilisateurs [DAH14].

Une solution consiste à proposer des systèmes hybrides, qui tirent parti des avantages des deux approches citées précédemment. Cette solution permet de combler les lacunes de l'une des approches sur des cas d'utilisation précis. Ces approches hybrides peuvent être implémentées de diverses façons :

- En combinant les résultats produits par chacune des deux approches exécutées indépendamment ;
- En sélectionnant un aspect précis d'une des approches et en l'intégrant à l'autre (compléter les approches) ;

- En unifiant les deux approches dans un modèle global.

Un grand nombre de systèmes de recommandation actuellement en exploitation fonctionne sur un modèle hybride. Parmi les systèmes hybrides les plus connus, le système de recommandation mis en place par le géant Américain de la vidéo à la demande sur Internet Netflix [DAH14]. Après avoir proposé un prix d'un million de Dollars en 2006 aux travaux obtenant les meilleures recommandations sur les données issues de la plateforme, l'entreprise a intégré les propositions les plus pertinentes dans sa version du système de recommandation mis en production. Il est reporté que plus des deux tiers du total des vidéos consommées sur cette plateforme sont issues des seules recommandations présentées aux utilisateurs ce qui reflète l'importance de la capacité à fournir des recommandations de qualité.

### **Exemple**

Comme illustré dans la figure 6. D'abord, le système applique à part une ou plusieurs techniques de filtrage de base telles que filtrage collaboratif, filtrage basé sur le contenu et filtrage démographique. Cette phase préliminaire génère les recommandations candidates, et qui sont combinées par certaines méthodes d'hybridation dans la phase de couplage, afin de produire les recommandations finales destinées à l'utilisateur.

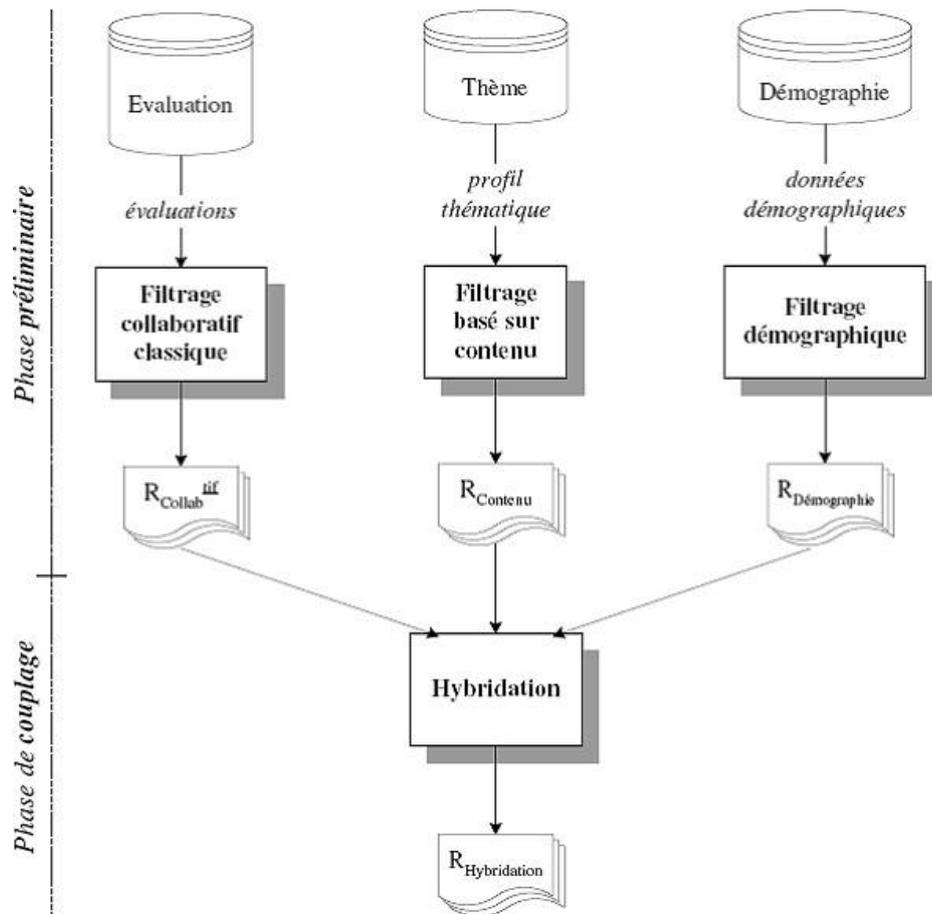


Figure 6. Hybridation classique de filtrage

## 6.5 Calcul de prédiction

La prédiction est l'opération qui consiste à estimer l'intérêt que portera un utilisateur pour un document qui serait recommandé pour lui.

### 6.5.1 Prédiction dans le filtrage cognitif

Dans le cadre du filtrage par le contenu, la prédiction de l'opinion de l'utilisateur sur un document donné, est estimée par un calcul de rapprochement entre les thèmes contenus dans le profil de l'utilisateur et les thèmes extraits des documents par indexation.

### 6.5.2 Prédiction dans le filtrage collaboratif

Quant au filtrage collaboratif, la prédiction de l'opinion de l'utilisateur sur un document se calcule par rapprochement entre les évaluations passées de l'utilisateur et des autres utilisateurs de la communauté, sur les mêmes documents.

Le filtrage collaboratif tient compte de la proximité entre les utilisateurs, afin de ne recommander un document qu'aux utilisateurs proches de celui qui l'a apprécié [LEE05]. Une des spécificités du filtrage collaboratif réside dans le fait que cette proximité qui contribue à la formation de communautés d'utilisateurs est estimée sur l'unique base de la liste de leurs évaluations passées sans tenir compte du contenu des documents.

Les systèmes de filtrage collaboratif peuvent être classés en deux grandes catégories : les algorithmes basés « mémoire » et les algorithmes basés « modèle ».

### **Algorithmes basés « mémoire »**

Ce type d'algorithme utilise toute la base de données des évaluations des utilisateurs pour effectuer le calcul des prédictions. Ces algorithmes fonctionnent donc au-dessus de la base de données entière pour faire des prévisions.

L'évaluation prédite sur un nouvel item pour l'utilisateur actif est une somme pondérée des évaluations des autres utilisateurs de son voisinage. Les détails de calcul des poids donnent lieu à des algorithmes différents tels que l'algorithme basé sur la corrélation qui calcule le poids comme la corrélation entre les utilisateurs, et l'algorithme basé sur la similarité de vecteurs dans lequel le poids est calculé comme un cosinus entre les vecteurs formés par les évaluations, des utilisateurs.

Dans notre étude, nous utiliserons la Corrélation de Pearson pour calculer la similarité et la prédiction des évaluations.

### **Corrélation de Pearson**

La fonction de similarité de Pearson est peut-être le meilleur algorithme connu pour les systèmes de recommandation basés sur l'utilisateur. Il a été introduit pour la première fois par le projet GroupLens en 1994 et utilisé depuis lors comme base de comparaison.

Lorsqu'on utilise la corrélation de Pearson, la similarité est représentée par une échelle de -1 à +1 où une valeur élevée positive suggère une forte corrélation, une valeur négative élevée suggère le contraire (quand on dit vrai, l'autre dit faux) et enfin, une corrélation nulle indique des échantillons non corrélés [LEV14].

L'équation de similarité de corrélation de Pearson basée sur l'utilisateur est définie comme suit :

$$s(u, v) = \frac{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{ui} - \bar{r}_u)^2 \sum_{i \in I_{uv}} (r_{vi} - \bar{r}_v)^2}} \quad (1)$$

Où  $(u, v)$  représente la similarité entre l'utilisateur  $u$  et  $v$ ,  $I_{uv}$  représente l'ensemble des éléments notés par les utilisateurs  $u$  et  $v$ ,  $r_{ui}$  et  $r_{vi}$  représentent la notation de l'utilisateur  $u$  et  $v$  pour l'item  $i$ , et  $\bar{r}_v$  et  $\bar{r}_u$  représentent les moyennes des évaluations des utilisateurs  $v$  et  $u$ , sur tous les items évalués.

Dans le cas des systèmes de recommandation basée sur les items, la formulation devient :

$$s(i, j) = \frac{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)(r_{uj} - \bar{r}_j)}{\sqrt{\sum_{u \in U_{ij}} (r_{ui} - \bar{r}_i)^2 \sum_{u \in U_{ij}} (r_{uj} - \bar{r}_j)^2}} \quad (2)$$

Où  $(i, j)$  représente la similarité entre l'utilisateur  $i$  et  $j$ ,  $U_{ij}$  représente l'ensemble des utilisateurs communs qui ont évalué les éléments  $i$  et  $j$ , et  $\bar{r}_i$  et  $\bar{r}_j$  représente la note moyenne des éléments, parmi tous les utilisateurs qui ont évalué l'élément.

Pour la prédiction de l'évaluation, la formule est comme suit :

$$\hat{r}_{ui} = \bar{r}_u + \frac{\sum_{v \in Ni(u)} w_{uv} (r_{vi} - \bar{r}_v)}{\sum_{v \in Ni(u)} |w_{uv}|} \quad (3)$$

Où  $Ni(u)$  représente le nombre de voisins ayant l'élément  $i$  en commun avec l'utilisateur  $u$ , dont  $v$  est un utilisateur voisin particulier, et  $w_{uv}$  est la similarité entre l'utilisateur  $u$  et l'un de ses voisins  $v$ , et  $\bar{r}_u$  représente la moyenne des évaluations de l'utilisateur  $u$ , sur tous les items évalués.

De même, pour les systèmes basés sur les items, l'équation précédente devient :

$$\hat{r}_{ui} = \bar{r}_i + \frac{\sum_{j \in Nu(i)} w_{ij}(r_{ju} - \bar{r}_j)}{\sum_{j \in Nu(i)} |w_{ij}|} \quad (4)$$

Où  $\hat{r}_{ui}$  représente l'évaluation prédite pour l'utilisateur  $i$ ,  $Ni(u)$  représente le nombre de voisins ayant l'élément  $i$  en commun avec l'utilisateur  $u$ , et  $w_{uv}$  est la similarité entre l'item  $i$  et l'un de ses voisins  $j$ , et  $\bar{r}_i$  et  $\bar{r}_j$  représente la note moyenne des éléments, parmi tous les utilisateurs qui ont évalué l'élément.

## 6.6 Autres formes de filtrage

### 6.6.1 Filtrage actif

Les limitations du filtrage collaboratif peuvent trouver des solutions dans le filtrage dit actif qui contribue à réduire le démarrage à froid par la possibilité offerte aux utilisateurs de la communauté de se recommander mutuellement des documents. Lorsqu'un utilisateur trouve des documents plus ou moins intéressants pour certains autres utilisateurs qu'il connaît, il peut les recommander pour eux [DEN04].

### 6.6.2 Filtrage différé

Le filtrage différé consiste à filtrer périodiquement les documents au fur et mesure de leur arrivée, pour cela le système stocke les documents dès leur arrivée et filtre chaque ensemble de documents stockés disponible à la fin de chaque période [TMA01].

### 6.6.3 Filtrage adaptatif

Il consiste à construire un premier modèle grâce à une requête formulée en langage naturel, puis à simuler un flux de documents. Le système peut tirer parti de la

pertinence ou de la, non-pertinence des documents sélectionnés pour s'améliorer au cours du temps [BIS03].

## 6.7 Systèmes de filtrages renommés

### 6.7.1 Système COCoFil

COCoFil (Community-Oriented Collaborative Filtering) est une plateforme ouverte d'essai et d'évaluation pour le filtrage collaboratif. Elle est orientée vers la conception de fonctionnalités pour une meilleure gestion de communautés d'utilisateurs. Elle a été conçue d'une manière modulaire pour permettre l'ajout d'éventuelles fonctionnalités, en plus du stockage systématique d'un maximum d'informations même celles qui sont que rarement exploitées [DEN04].

Elle offre le moyen de paramétrer certaines fonctionnalités pour qu'elles s'adaptent aux besoins de l'utilisateur du système, comme la personnalisation de l'interface, l'activation ou la désactivation de certaines fonctions et prend en charge les fonctions standards relatives à l'interaction des utilisateurs comme le calcul de prédiction et d'autres fonctions orientés plutôt vers les communautés d'utilisateurs pour en tirer le meilleur bénéfice des possibles relations entre utilisateurs du système.

Les principales fonctionnalités offertes à l'utilisateur sont :

**Perception des autres :** Chaque utilisateur peut consulter la liste des personnes qui ont des profils similaires au sien, et ainsi connaître de nouvelles personnes ayant les mêmes centres d'intérêt que lui.

**Carnet d'adresses :** Chaque utilisateur possède un carnet d'adresses qui enregistre des informations utiles des personnes « intéressantes » avec lesquelles il souhaite échanger régulièrement des recommandations.

**Consultation du profil :** La plateforme permet aux utilisateurs de consulter leur profil. Un profil est donc une liste de tuples composés de l'identificateur de l'objet, la valeur de l'évaluation personnelle, et la date de cette évaluation.

**Gestion des documents et recommandations :** l'utilisateur a la possibilité d'organiser ses documents selon une structure arborescente de répertoires.

**Paramétrage utilisateur :** COCoFil permet à l'utilisateur de configurer certains aspects pour personnaliser son utilisation de la plateforme tels que la visualisation choisir la façon d'afficher les contenus dans ses répertoires personnels, son carnet d'adresses, son profil, et les listes de personnes de profils similaires et

l'évaluation en choisissant le niveau de confidentialité, le filtrage collaboratif pour décider si une évaluation doit être prise en compte dans le profil et le niveau d'exploitation pour le choix du régime débutant ou expert.

### **6.7.2 Système de Maltz et Elrich**

Le système de Maltz et Elrich est conçu sur l'idée que les utilisateurs recherchant l'information devraient pouvoir se servir de ce que d'autres ont déjà trouvé et évalué. Pour ce faire les utilisateurs font recours à leurs boîtes Email pour envoyer des pointeurs ou des liens URLs vers les documents intéressants des amis et des collègues. Cependant cette pratique requiert des efforts de la part de l'expéditeur qui peut se lasser d'un bout d'un temps ou oublier carrément d'envoyer les informations aux personnes concernées.

L'idée de base du système de Maltz est d'offrir à l'utilisateur un moyen commode et simple autre que le mail pour qu'il envoie les pointeurs aux documents intéressants aux autres de la communauté sans pour autant interrompre sa session de recherche d'informations. De plus, tous les échanges entre utilisateurs de la communauté sont stockés dans une base qui servira de références pour des besoins ultérieurs [MAL95].

### **6.7.3 Système Tapestry**

Tapestry, fût l'un des premiers systèmes de filtrage, utilisant le principe du filtrage collaboratif. Il a été conçu en vue de la gestion des e-mails et dans lequel l'utilisateur n'est pas limité à fournir de simples jugements d'acceptation ou de rejet, mais peut faire des annotations en texte libre du genre « J'ai bien aimé » ou « Je déteste », ainsi les utilisateurs peuvent transmettre des jugements sur la valeur des documents qu'ils lisent.

Les autres utilisateurs peuvent alors opérer des recherches parmi ces documents non seulement sur la base de leur contenu, mais également sur la base des jugements qu'ont portés d'autres utilisateurs à leur sujet [BER03].

### **6.7.4 Système ReferralWeb**

Ce système offre une autre manière d'aborder la recherche de l'information et ce en valorisant les relations personnelles entre les producteurs de documents et les chercheurs d'informations. Contrairement aux autres systèmes qui favorisent l'anonymat, le système ReferralWeb est basé sur la connaissance des interlocuteurs. Son but en adoptant cette stratégie est de focaliser efficacement l'information dans une session de recherche. Le système ReferralWeb est orienté

vers plus vers la découverte de réseaux sociaux existants plutôt que la construction de nouvelles communautés.

Un réseau social est alors modélisé par un graphe où les nœuds représentent les personnes et les arcs les relations directes entre personnes. Plusieurs méthodes peuvent être utilisées pour déterminer ces relations :

- Soit, on demande à l'utilisateur d'introduire la liste de ses collègues proches
- Soit, on analyse les entêtes des Emails de l'utilisateur, mais cette solution n'est exempte de problème de sécurité et de confidentialité.

### **Conclusion**

Le filtrage d'information est un domaine de recherche en plein essor, visant principalement à extraire pour le compte de l'utilisateur, des flots de documents entrants, les informations susceptibles de l'intéresser.

Ces informations sont connues du système à travers la description du profil utilisateur qui devra être maintenu à jour pour refléter ses vrais centres d'intérêts. De plus en travaillant en collaboration avec d'autres dans le cadre du filtrage collaboratif, ce dernier trouvera un meilleur gain par le fait qu'il peut bénéficier de l'assistance, des conseils d'autres utilisateurs avec qui, il partage les mêmes préférences, sans perdre de vue que cette catégorie de filtrage résout un certain nombre de problèmes tel que, le filtrage des données multimédias par exemple.

---

## **Chapitre 2 : Prédiction des liens**

---

## 1. Introduction

Les systèmes de recommandation sont largement utilisés dans de nombreuses d'application pour suggérer des produits, des services et des informations aux utilisateurs potentiels. Les technologies de recommandation reposent sur des algorithmes de recommandation qui prennent les attributs utilisateur / article et les interactions utilisateur / article (évaluations, activités de navigation dans les articles, activités d'emprunt de livres, etc.) comme entrées pour prédire le niveau de correspondance non observé d'une paire utilisateur / article particulière.

Le filtrage collaboratif est l'un des algorithmes de recommandation les plus aboutis et les mieux étudiés, qui ne repose que sur les données d'interaction utilisateur-élément pour formuler des recommandations. Un algorithme de filtrage collaboratif standard basé sur l'utilisateur dérive d'abord les communautés d'utilisateur en identifiant les utilisateurs similaires en fonction de leurs interactions qui se chevauchent ou évaluations similaires d'éléments communs [HUA04]. Il formule ensuite des recommandations basées sur les expériences des voisins de l'utilisateur.

En dépit de son succès, le filtrage collaboratif est fortement limité par le problème de rareté des données, dans lequel les transactions d'éléments utilisateur fragmentées ne prennent pas en charge l'inférence valide des voisins de l'utilisateur.

Pour les recommandations basées sur des données de transaction binaires (les exemples dans les bibliothèques numériques incluent l'emprunt de livres et les activités de navigation dans les articles), nous avons proposé d'atténuer le problème de rareté en représentant les données de transaction en tant que liens dans un graphe bipartite contenant des nœuds utilisateur/élément. Sous cette représentation graphique, le problème de recommandation peut être considéré comme une tâche de sélection de liens non observés pour chaque nœud d'utilisateur et peut donc être modélisé comme un problème de prédiction de lien.

Dans cette étude, nous explorons les approches de prévision de lien développées récemment dans la modélisation de réseau et la littérature de prévision pour la recommandation basée sur les transactions.

## 2. Profil Utilisateur

### 2.1 Définition

Dans [MIR04], l'auteur divise les profils utilisateurs dans deux groupes : les profils qui représentent les préférences de l'utilisateur et ceux qui représentent ses intérêts. Par

exemple, dans [SCH03] les auteurs présentent une architecture pour la recherche d'information sur le Web qui utilise des profils représentant les préférences de l'utilisateur. Ils définissent : « un profil utilisateur est un ensemble de préférences concernant le comportement d'un moteur de recherche ainsi que les contraintes sur les résultats qu'il présente à l'utilisateur ». Les préférences et contraintes dans ce cas concernent le choix de présentation de résultats (affichage au maximum de 25 documents par page), le format de documents (PDF, HTML ...), le format structurel de documents (abstracts ou document entier) etc. Les profils représentant les intérêts de l'utilisateur sont plus répandus que ceux qui représentent les préférences. De plus, ils concernent plus d'aspects, aux niveaux de représentation, de construction, et d'utilisation.

### 2.2 Profils à court-terme et à long-terme

Nous avons différents intérêts à différents moments. Par exemple, un chercheur dans le domaine de la recherche d'information s'intéresse généralement aux documents parlant de la recherche d'information. Cependant, à un moment spécifique, il veut chercher des informations concernant les prix de billets d'avion pour aller à une conférence. Ainsi la « recherche d'information » est son intérêt à long-terme et le « prix des billets d'avion » est son intérêt à court-terme. A partir de cet exemple, nous pouvons dire que les notions de profils à court-terme et profils à long-terme [KAZ04, HUN05, DAN99] décrivent respectivement les préférences éphémères et persistantes d'utilisateurs. D'autres auteurs [PHI03] utilisent ces notions pour décrire les préférences spécifiques et générales d'utilisateurs bien que « éphémère ne correspond pas toujours avec spécifique et « persistant » ne correspond pas toujours avec « général ». Normalement, le profil à court-terme est plus important que le profil à long-terme parce qu'il décrit mieux l'intérêt de l'utilisateur au moment présent.

### 2.3 Modèles de Représentation du profil utilisateur

#### 2.3.1 Modèle vectoriel

Dans le modèle vectoriel, chaque profil utilisateur se compose d'un ou plusieurs vecteurs de termes. Chaque terme est associé avec une valeur. Ce modèle est le modèle le plus utilisé. En utilisant ce modèle, nous pouvons calculer facilement la similarité cosinus d'un document quelconque avec le profil d'un utilisateur si le document est aussi représenté par un vecteur de termes. Dans un cas particulier de ce modèle, les poids des termes dans le vecteur n'est plus un nombre réel mais une valeur booléenne [DAN99, PAZ96]. Cette valeur représente la présence ou non d'un terme dans le profil.

### 2.3.2 Modèle sémantique à base d'ontologie

Un autre modèle populaire de représentation de profils utilisateurs est le modèle sémantique à base d'ontologie. Dans ce modèle, un profil est une hiérarchie de concepts pondérés. Chaque nœud dans la hiérarchie est un concept. Le poids attaché avec un concept représente l'intérêt de l'utilisateur avec ce concept. Ce poids peut être changé pour mettre à jour l'intérêt de l'utilisateur. De plus, chaque concept est souvent représenté par un vecteur de termes pondérés. Le poids attaché avec un concept représente l'intérêt de l'utilisateur tandis que ce vecteur représente le « contenu » de ce concept. Ce vecteur peut être construit à partir d'un ensemble de documents assignés à ce concept.

L'utilisation du modèle ontologie pour représenter les profils utilisateurs peut aider à mieux connaître les intérêts des utilisateurs par rapport au modèle vectoriel. Par exemple, si on représente un profil utilisateur par un vecteur de termes pondérés et dans ce vecteur contient un terme « instrument », on ne sait pas exactement si ce terme concerne des instruments musiques ou les autres types d'instruments. Cependant, si on utilise le modèle ontologie (comme illustré dans la figure 4), on peut facilement régler ces ambiguïtés : dans cet exemple, le concept « instruments » est un concept « fils » du concept « music », alors il s'agit des instruments musiques.

De plus, un autre avantage du modèle ontologie est qu'on peut propager la valeur d'intérêt d'un concept vers les autres concepts reliés (par exemple, son concept « père ») afin de trouver des nouveaux centres d'intérêt [PAZ96].

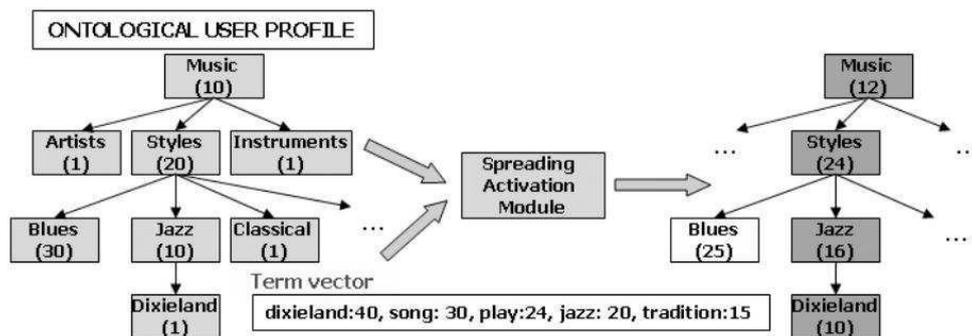


Figure 7. Exemple du profil utilisateur représenté par le modèle d'ontologie.

### 2.3.3 Modèle Multidimensionnel

Le travail de Amato et al [AMA11]. est un des premiers travaux vers la construction d'un modèle multidimensionnel pour représenter des profils utilisateurs. Cette représentation donne une description globale des utilisateurs en prenant en compte plusieurs dimensions différentes. Les informations concernant les utilisateurs peuvent être classifiées dans cinq catégories différentes, chaque catégorie est une dimension :

- La catégorie des données personnelles contient des données d'identification personnelles de l'utilisateur (nom, date de naissance, contact).
- La catégorie de recherche contient des préférences et des restrictions sur les documents que l'utilisateur est en train de rechercher.
- La catégorie de livraison sont des spécifications concernant le mode de livraison des informations trouvées (courriel, fax, Web, temps de livraison etc.).
- La catégorie de données des actions contient des enregistrements sur l'interaction de l'utilisateur avec le système de recherche et les données de navigation (pages Web visités, documents lus, jugements de pertinence etc.).
- Enfin la catégorie de données de sécurité est une collection des préférences de l'utilisateur concernant des conditions d'accès aux informations du profil utilisateur.

### 2.3.4 Modèle matrice utilisateurs-items

Un autre modèle souvent utilisé dans les systèmes de recommandation collaborative est la représentation par une matrice Utilisateurs-Items. Chaque ligne de la matrice représente un utilisateur et chaque colonne représente un item. Une cellule  $[i, j]$  de la matrice contient le vote de l'utilisateur  $i$  pour l'item  $j$  (ou rien si utilisateur n'a pas voté pour cet item). Dans ce modèle, le profil d'un utilisateur est considéré comme un vecteur de votes.

## 2.4 Acquisition d'Information

Actuellement, il existe trois méthodes principales pour acquérir ces informations : la méthode d'acquisition implicite, la méthode d'acquisition explicite et la méthode hybride.

### 2.4.1 Acquisition explicite

Dans cette méthode, les utilisateurs doivent fournir explicitement les informations nécessaires au système. Une telle approche a été utilisée depuis longtemps dans plusieurs systèmes de recherche d'information (par exemple la méthode de retour de pertinence de Rocchio [ROC71]). Il existe plusieurs formes d'acquisition explicite :

**Entrée directe par les utilisateurs :** Les utilisateurs doivent entrer leurs intérêts sous la forme de mots clés. Malgré sa simplicité apparente, c'est une approche qui demande beaucoup d'effort du côté des utilisateurs pour clarifier leurs intérêts.

**Classement « binaire » par les utilisateurs :** Les utilisateurs doivent classer les items (pages Web, livres . . .) dans deux classes « intéressants » ou « inintéressants ».

**Vote :** Les utilisateurs doivent donner des votes explicites pour mesurer leurs intérêts avec les items qu'ils doivent évaluer. Par exemple, dans le système Syskill & Webert , les pages Web sont évaluées selon trois niveaux hot, lukewarm, cold [MIQ03].

**Stéréotype :** Quelques systèmes demandent aux utilisateurs de répondre à un ensemble de questions prédéfinies et puis ils utilisent ces informations pour déduire leurs profils.

### 2.4.2 Acquisition implicite

Dans cette approche, le système acquiert les informations nécessaires pour construire les profils utilisateurs implicitement en surveillant ou traçant leurs actions. Par exemple, si un utilisateur sauvegarde un document sur son disque, il est probablement intéressé par ce document et il a l'intention de l'utiliser dans l'avenir.

De manière similaire, si l'utilisateur utilise beaucoup de temps pour lire un document Web et/ou suit beaucoup de liens dans ce document, c'est aussi une preuve qu'il s'intéresse à ce document. Dans [LET95], le temps utilisé et les liens suivis dans un document Web ont été utilisés comme des preuves de l'intérêt de l'utilisateur. Une autre approche très populaire est de regarder les historiques de navigation Web de l'utilisateur pour trouver les types de documents qui sont intéressants pour lui [KAZ04].

Dans [THO05], les auteurs présentent une étude concernant l'utilisation des clics de souris (clickthrough) comme les informations implicites.

### 2.4.3 Approche hybride

Quelques systèmes ont choisi de combiner les deux précédentes méthodes pour obtenir une meilleure performance. Dans [ZHA00], un profil utilisateur contient des termes pondérés, chaque fois qu'un document est jugé pertinent, le poids d'un terme dans son profil est mis à jour en utilisant les paramètres suivants : le vote explicite, le temps utilisé pour lire ce document, le nombre de liens suivis et l'action sauvegardée dans les signets de ce document.

Dans [MIQ03], les auteurs font une liste de 37 systèmes qui utilisent différentes approches d'acquisitions d'information d'utilisateurs. Parmi eux, 20 systèmes utilisent des approches explicites, 8 systèmes utilisent des approches implicites, les approches hybrides sont utilisées par 9 autres systèmes.

### 2.5 Mise à jour de profils utilisateurs

#### Technique TF-IDF

Dans les systèmes personnalisés, la méthode la plus utilisée pour construire des profils utilisateurs est la technique **tf-idf** et ses variantes. C'est une technique issue du domaine de la recherche d'information pour la pondération de termes dans le modèle vectoriel, comme le modèle le plus utilisé pour représenter des profils utilisateurs.

Quelques systèmes utilisant cette approche [LIR98, DAN99]. Dans [LIR98], un profil se compose de  $N$  vecteurs de termes pondérés. Chaque vecteur représente un domaine d'intérêt de l'utilisateur, chaque fois qu'un document est jugé pertinent par l'utilisateur, le système construit le vecteur **tf-idf** de ce document, après cette étape on obtient un ensemble de  $N + 1$  vecteurs pondérés ( $N$  vecteurs profils et 1 nouveau vecteur de document). Puis le système calcule la similarité cosinus entre chaque paire de vecteurs dans cet ensemble et combine les deux vecteurs les plus similaires. Dans cette approche, le profil est mis à jour incrémentalement chaque jour.

Dans le système NewsDude [DAN99], le profil à court-terme d'utilisateur se compose de plusieurs documents pour lesquels il a voté, chaque document est représenté par son vecteur **tf-idf**. Chaque fois qu'un nouveau document arrive, le système va d'abord extraire le vecteur **tf-idf** de ce document. Puis il compare la similarité cosinus de ce document avec les autres documents dans le corpus : les documents ayant une similarité avec le nouveau document plus élevée qu'un seuil prédéfini seront filtrés (ou sélectionnés). La prévision de vote du nouveau document sera la valeur moyenne de tous les votes que l'utilisateur a effectué pour les documents filtrés.

Le système utilise cette prévision de vote pour décider de recommander le nouveau document à l'utilisateur ou non.

### 3. Définition de Contexte

Le contexte est n'importe quelle information qui peut être utilisée pour caractériser une situation d'une entité. Une entité peut être une personne, un endroit, ou un objet que l'on considère comme étant pertinent à l'interaction entre un utilisateur et une application y compris ces deux dernier.

Le contexte peut être un ensemble de circonstances liées, situation où un phénomène apparait, un événement se produit. En informatique, le contexte est l'ensemble d'informations concernant un événement en rapport principalement avec sa localisation, qui permet au système informatique de fonctionner et de s'adapter en conséquence.

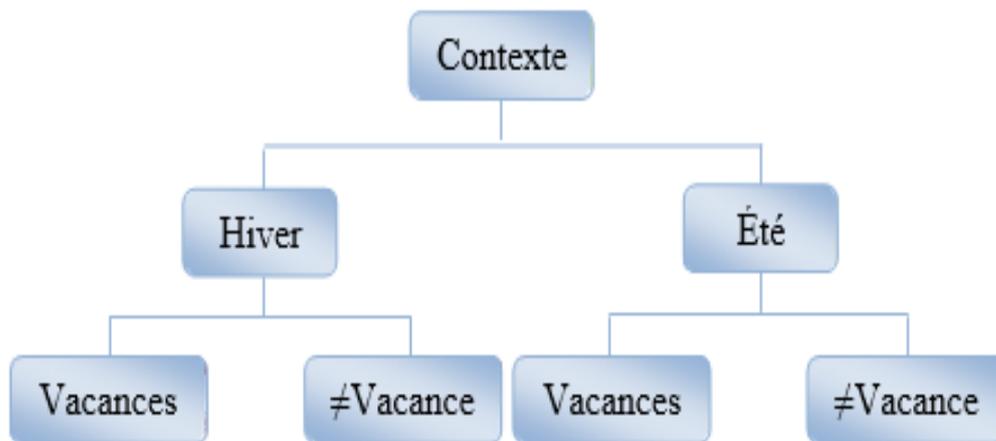


Figure 8. Modélisation Contextuelle

### 4. Recommandation par Contexte

Dans la plupart des systèmes de recommandation personnalisés, certains problèmes peuvent survenir, comme par exemple, recommander des produits inappropriés à un utilisateur, qui effectue un achat d'un cadeau sur internet pour l'offrir à son ami. Le système risque de lui recommander des produits similaires, alors qu'il ne s'agit pas de ses propres préférences.

Ce problème peut être surmonté si le système connaît " le contexte " dans lequel la transaction a eu lieu. Bien qu'il ne soit pas évident d'obtenir ce genre d'information, il est possible de la déduire des données dont on dispose d'une façon explicite (demande à l'utilisateur de spécifier les variables contextuelles) ou implicite (comme par exemple remarquer si le lieu de résidence d'un utilisateur a

changé, ou bien, quand est-ce qu'il a effectué les transactions à travers ses fichiers journaux).

D'après S. Lombardi et al. [LOM09], le modèle contextuel donne toujours de meilleurs résultats par rapport au modèle non-contextuel en termes de précision (nombre de cas correctement classés parmi tous les cas), et de rappel (nombre de tous les cas estimés pertinent par l'utilisateur et qui sont prévus correctement).

## 5. Dimensions du contexte

M. Daoud [DAO07], a proposé dans le cadre de son travail deux dimensions principales du contexte de recherche (le contexte de l'utilisateur et le contexte de la requête). Elle vise à intégrer ces deux dimensions du contexte dans un processus de recherche d'information (ou recommandation) afin d'asseoir un système de recherche d'information contextuel. La première permet de caractériser l'utilisateur à travers ses centres d'intérêts à court et à long terme et ses préférences. La seconde dimension est liée à la requête et permet de cibler la nature des résultats attendus par l'utilisateur.

### 5.1 Contexte lié à l'utilisateur

On définit le contexte lié à l'utilisateur par des éléments qui peuvent être donnés explicitement par l'utilisateur ou déduits implicitement par le système. On cite les centres d'intérêts qui peuvent appartenir aux domaines de la science, sport, humanité, etc., et les préférences liées à la qualité des informations attendues. Ces préférences sont données explicitement et comportent les deux aspects suivants :

- La fraîcheur : l'utilisateur peut s'intéresser à l'information la plus récente ou bien à des documents liés à des dates bien précises,
- La granularité : l'utilisateur peut s'intéresser à un certain niveau de détail de la réponse attendue, cela peut déterminer la structure de la réponse retournée : un document intégral ou une partie de document (passage), une page d'accueil ou une page cible.

### 5.2 Contexte lié à la requête

On définit ce contexte par différents paramètres descriptifs et éventuellement mesurables à partir de la requête ou à partir du profil des  $n$  top documents retournés par la requête. Soient les paramètres suivants :

- Clarté de la requête : ce paramètre quantifie le degré d'ambiguïté de la requête et permet de mieux cibler la recherche,
- Degré de couverture de la requête par les services web : ce paramètre permet d'orienter la réponse retournée vers un document ou service.

## 6. Emergence du contexte

L'incorporation de l'information contextuelle peut se faire à différents étapes du processus dans un système de recommandation. [ADO11] Adomavicius, G. and Tuzhilin définissent trois grandes approches de contextualisation suivant le moment où le contexte est injecté. Ces approches sont les suivantes :

### 6.1 Pré-filtrage contextuel

L'incorporation du contexte par pré-filtrage ou prétraitement consiste à sélectionner un sous-ensemble de données significatif pour le contexte dans lequel on se situe et de restreindre le processus de recommandation à ce sous-ensemble. Ceci implique donc de construire un modèle pour chaque contexte [BEN17]. Pour illustrer cette approche prenons l'exemple d'un système de recommandation de films qui utilise le contexte temporel : si un utilisateur souhaite regarder un film pendant le weekend, seuls les films disponibles pendant le weekend sont candidats à la recommandation et seules les notes des utilisateurs ayant vu les films pendant le weekend sont utilisées pour la prédiction de notes. L'utilisation de ce filtrage a priori a été critiquée, car l'ensemble des données est réduit et peut créer des problèmes pour la prédiction de notes si le système ne dispose pas d'assez de données.

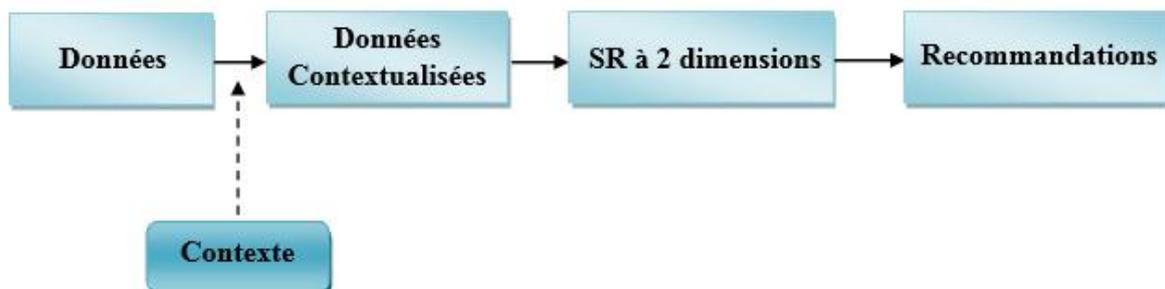


Figure 9. Pré-Filtrage Contextuel

## 6.2 Post-filtrage contextuel

Dans une approche d'incorporation du contexte par post-filtrage contextuel, le système de recommandation ne prend pas en compte les données contextuelles lors du processus de recommandation. Les sorties des algorithmes de recommandation sont modifiées a posteriori pour réordonner la liste des items recommandés en fonction du contexte. Par exemple, un système de recommandation de lieux touristiques utilisera la situation géographique de l'utilisateur (contexte de localisation), et peut décider d'éliminer a posteriori les recommandations de lieux trop éloignés de la position de l'utilisateur [BEN17].

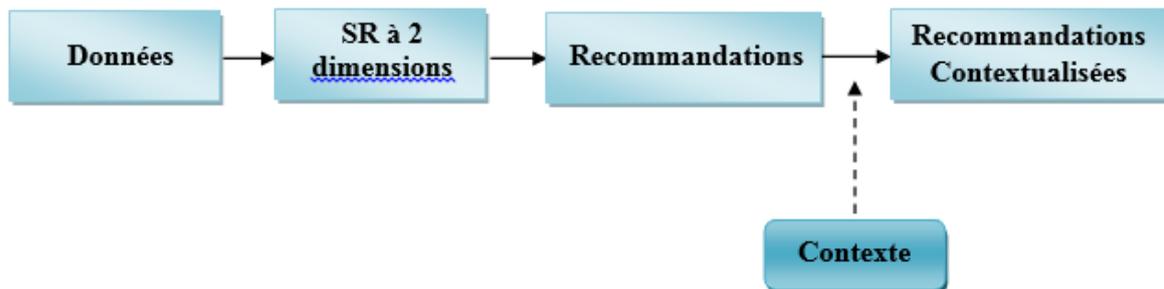


Figure 10. Post-Filtrage Contextuel

## 6.3 Modélisation directe du contexte

L'approche de modélisation du contexte consiste à intégrer directement les informations contextuelles dans le processus de recommandation pour la prédiction de notes pour les items. Pour intégrer le contexte, [KAR10] proposent des méthodes de factorisation. Pour ces méthodes, en plus des deux premières dimensions traditionnellement utilisées pour les items et les utilisateurs, chaque type de contexte est considéré comme une nouvelle dimension. La note n'est plus considérée comme une fonction avec les deux paramètres item et utilisateur, mais une fonction avec comme paramètres l'item, l'utilisateur et les aspects du contexte.

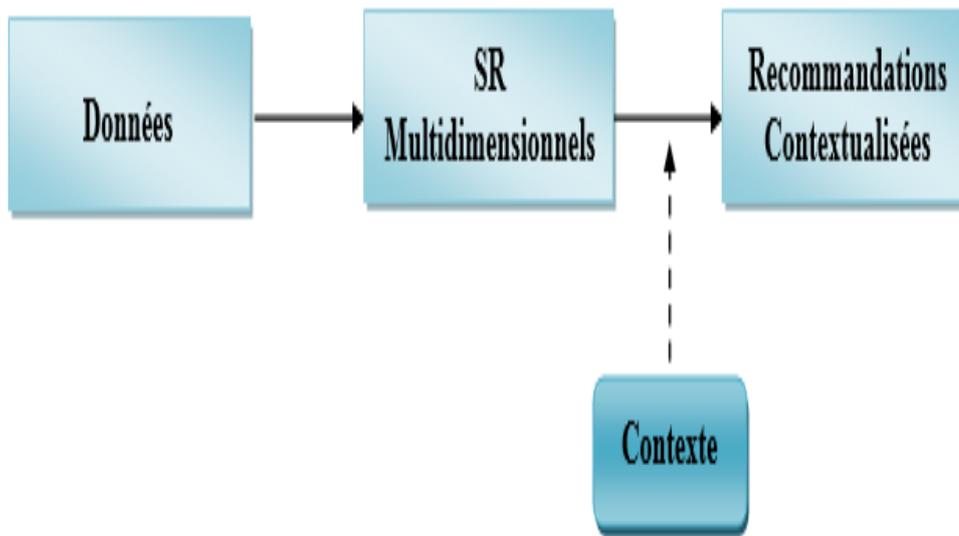


Figure 11. Modélisation Contextuelle

## 7. Proposition de recommandation hybride à base de prédiction des liens

### 7.1 Les Graphes

#### 7.1.1 Définition

Un graphe est un couple  $G = (V, E)$ , où  $V$  est un ensemble dont les éléments sont appelés sommets, et  $E$  est un ensemble de deux ensembles de sommets, dont les éléments sont appelés des arêtes (parfois des liens ou des lignes). Dans l'arête  $\{x, y\}$ , les sommets  $x$  et  $y$  sont appelés les extrémités ou les sommets extrêmes de l'arête. On dit que l'arête joint  $x$  et  $y$  et est incidente sur  $x$  et sur  $y$ . Un sommet peut exister dans un graphe sans arête incidente. Une boucle est une arête qui joint un sommet à lui-même. Les arêtes multiples sont deux arêtes ou plus qui joignent les mêmes deux sommets.

L'ordre d'un graphe  $|V|$  est son nombre de sommets. La taille d'un graphe est  $|E|$ , son nombre d'arêtes. Le degré ou la valence d'un sommet est le nombre d'arêtes incidentes à ce sommet, où une boucle compte double.

#### 7.1.2 Types de graphes

- **Graphe orienté**

Un graphe orienté est un graphe dans lequel les arêtes possèdent une orientation.

- $V$  un ensemble de sommets (aussi appelés nœuds ou points) ;

- $A \subseteq \{(x, y) \mid (x, y) \in V^2 \wedge x \neq y\}$  un ensemble de flèches (aussi appelées arêtes orientées — parfois simplement arêtes avec l'ensemble correspondant nommé  $E$  au lieu de  $A$  —, liens orientés ou lignes orientées), qui sont des couples de sommets distincts (c.-à-d. une flèche est associée à deux sommets distincts).

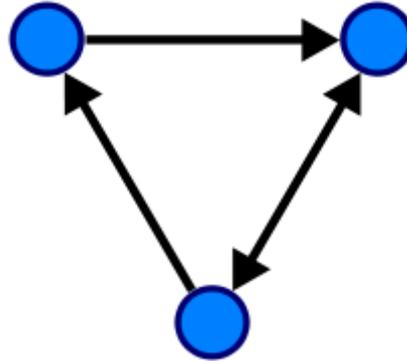


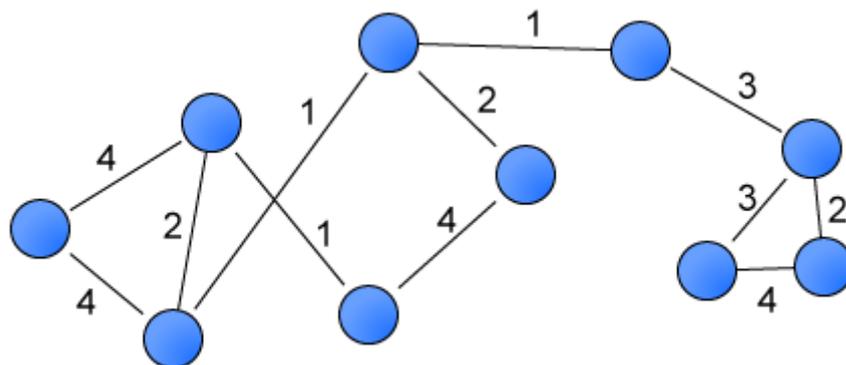
Figure 12. Un graphe orienté avec trois sommets et quatre arêtes.

- **Graphe mixte**

Un graphe mixte est un graphe composé d'arêtes non orientées et d'arêtes orientées. C'est un triplé  $G = (V, E, A)$  pour un graphe mixte simple et  $G = (V, E, A, \phi E, \phi A)$  pour un multigraphe mixte avec  $V, E, A, \phi E$  et  $\phi A$  définis comme précédemment. Les graphes orientés et non orientés en sont des cas particuliers.

- **Graphe pondéré**

Un graphe pondéré ou un réseau est un graphe où chaque arête porte un nombre (son poids). Ces poids peuvent représenter par exemple des coûts, des longueurs ou des capacités, en fonction du problème traité. Ces graphes sont fréquents dans divers contextes, comme l'évaluation qu'un utilisateur a donné à un item.



**Figure 13. Un graphe pondéré avec dix sommets et douze arêtes.**

- **Grappe biparti**

En théorie des graphes, un graphe est dit biparti s'il existe une partition de son ensemble de sommets en deux sous-ensembles U et V telle que chaque arête ait une extrémité dans U et l'autre dans V. Un graphe biparti permet notamment de représenter une relation binaire.

## 7.2 Prédiction des liens dans les graphes bipartis

Nous représentons les interactions utilisateur-item sous forme de liens dans un graphe  $G$  de type élément-item biparti. Le poids des arêtes est directement l'évaluation qu'un utilisateur a donné à un produit. Sur la base de la topologie de  $G$ , nous calculons certaines mesures de liaison  $w(u, i)$  pour chaque paire utilisateur-item non connecté  $\langle u, i \rangle$ . Ces mesures servent ensuite de points de repère pour évaluer la possibilité d'un lien reliant  $u$  et  $i$  en vue de la recommandation.

La plupart des mesures de liaison existantes sont proposées pour les graphes unipartis, les liaisons étant autorisées entre toute paire de nœuds. Nous avons révisé certaines mesures pour qu'elles aient un sens dans un graph biparti. Pour un nœud  $x$ , nous définissons  $\Gamma(x)$  comme l'ensemble des voisins de  $x$  dans  $N_h$ . Nous définissons également  $\hat{\Gamma}(x) = \bigcap_{c \in \Gamma(x)} \Gamma(c)$  comme l'ensemble des voisins des voisins de  $x$ .

**7.2.1 Common Neighbors :** Dans un graphe unipartite, le nombre de voisins communs de  $x$  et  $y$   $|\Gamma(x) \cap \Gamma(y)|$  pourrait représenter la similarité des nœuds. Dans notre graphe utilisateur-item, un nœud d'utilisateur est uniquement connecté à des nœuds d'élément. Ainsi,  $|\Gamma(x) \cap \Gamma(y)|$  sera toujours égal à  $\emptyset$  si  $x$  et  $y$  forment une paire utilisateur-item. La version bipartite de la mesure du voisin commun dans notre étude est :

$$|\Gamma(x) \cap \hat{\Gamma}(y)|.$$

**7.2.2 Coefficient de Jaccard :** Il mesure le nombre de voisins de  $x$  et  $y$  par rapport au nombre de nœuds voisins de  $x$  ou de  $y$ . Dans notre contexte, cette mesure est adaptée à :

$$\frac{|\Gamma(x) \cap \hat{\Gamma}(y)|}{|\Gamma(x) \cup \hat{\Gamma}(y)|}.$$

**7.2.3 Adamic/Adar :** Calcule les voisins communs par les objets et définit la similitude entre eux :

$$\sum_{z \in \Gamma(x) \cap \hat{\Gamma}(y)} \frac{1}{\log |\Gamma(z)|}$$

**7.2.4 Rattachement Préférentiel :**  $|\Gamma(x)| \cdot |\Gamma(y)|$  cette mesure relie l'interaction future avec la popularité de l'élément et le niveau d'activité des utilisateurs dans le système.

Les trois premières mesures basées sur le voisinage impliquent une intersection entre des ensembles de voisins, qui sont essentiellement similaires aux algorithmes de filtrage collaboratif standard basés sur le voisinage. La mesure de rattachement préférentiel a ses racines dans les modèles de réseau sans échelle et est liée au mécanisme de croissance du réseau.

- **Projection pondérée d'un graphe biparti**

Cette approche a été présentée par Zhou et al. [ZHO07], mais dans ce cas, nous utiliserons un modèle pondéré du graphe bipartite et un processus d'allocation pondérée des ressources. Supposons qu'une certaine quantité de ressources (par exemple, un pouvoir de recommandation) soit associée à chaque utilisateur et que le poids  $rp(u, v)$  représente la proportion de la ressource que l'utilisateur  $u$  veut distribuer vers  $v$ .

Il existe deux étapes dans le processus d'allocation de ressources :

Premièrement, chaque utilisateur distribue sa ressource initiale par rapport à ses préférences pour les items qu'il a évalué, chaque produit reçoit une ressource des utilisateurs qui l'ont évalué. Ensuite, le produit  $p$  reçoit  $r_{u,p}/R_u$ ,

Où  $R_u = \sum_{q \in P} (r_{u,q})$  est le nombre de produits que l'utilisateur  $u$  a évalué.

Deuxièmement, l'utilisateur  $u$  reçoit des ressources de la part de tous les produits qu'il a évalué  $r_{u,p}/R_p$ , Où  $R_p = \sum_{v \in U} (r_{v,q})$  est le nombre d'utilisateurs qui ont évalué  $p$ .

Ainsi, le poids (ou score de similarité) peut être exprimée par la formule suivante :

$$rp(u, v) = \sum_{p \in P} \frac{r_{v,p}}{R_p} \frac{r_{u,p}}{R_u} = \frac{1}{R} \sum_{p \in P} \frac{r_{u,p} r_{v,p}}{R_p} \quad (5)$$

Une fois que le pouvoir de recommandation entre utilisateurs est calculé, nous pouvons l'utiliser pour générer une recommandation personnelle, similaire à la procédure d'utilisation de la similarité pour la prédiction dans des algorithmes de recommandation collaborative standards. De plus, si toutes les recommandations des utilisateurs sont prises en compte, puisque  $r_p$  est normalisé avec l'utilisateur, la prédiction de l'utilisateur  $u$  sur le produit  $p$  peut être formulée ainsi :

$$r_{c,p} = \frac{\sum_{v \in U} rp(v,c)r_{v,p}}{\sum_{v \in U} rp(v,c)} = \sum_{v \in U} rp(v,c)r_{v,p} \quad (6)$$

Prenons l'exemple du Tableau 1, la partie gauche est la matrice d'évaluation user-item, celle du milieu est la matrice des Power Recommendations, et la partie de droite est la matrice de prédiction.

**Tableau 2. Similarité dans un graphe biparti**

	Rating				Recommandation Power			Projection-based prediction			
	P1	P2	P3	P4	C1	C2	C3	P1	P2	P3	P4
C1	3	2		1	0.78	0.11	0.11	2.33	1.72	0.72	1.2
C2		1	3		0.17	0.65	0.19	0.33	0.87	2.19	0.61
C3			1	2	0.22	0.25	0.53	0.33	0.41	1.09	1.17

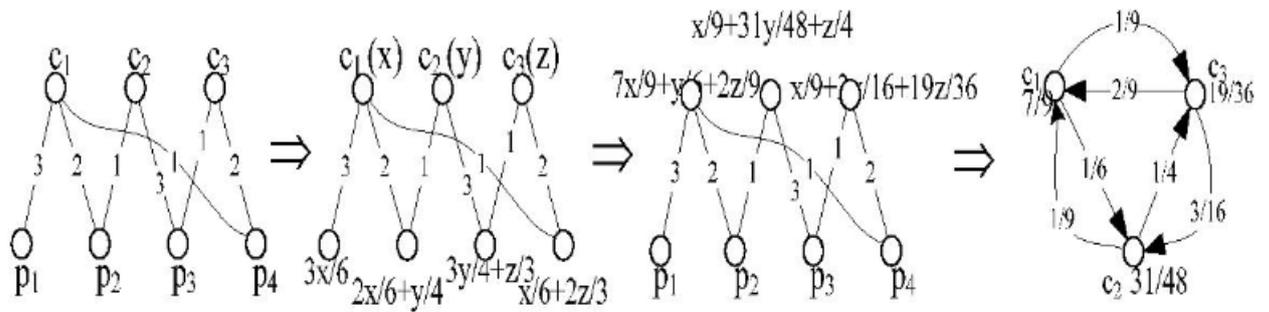


Figure 14. Projection de graphe biparti user-item

## 8. Conclusion

Dans ce travail, nous avons exploité diverses formules de prédiction de liens et étudié leur application aux graphes bipartis. Nous avons constaté que les métriques de similarité de prédiction de liaison couramment utilisées pourraient également être efficaces sur les réseaux bipartis si elles étaient modifiées en les appliquant sur un plus grand ensemble de codes fermés que de voisins. Nous avons exploré diverses métriques de similarité et avons constaté que les méthodes de similarité qui fonctionnent bien sur des réseaux homogènes fonctionnent également bien sur des réseaux bipartis après modification. Nous avons aussi présenté un algorithme de recommandation personnel qui utilise la projection d'un réseau pondéré d'items-utilisateurs, ce qui surpasse les mesures de similarité connues comme la corrélation de Pearson ou la mesure de cosinus.



---

# **Chapitre 3 : Expérimentations Et Résultats**

---

## 1. Environnement de travail

### 1.1 Java

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au SunWorld.

La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels que UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java.

Le langage reprend en grande partie la syntaxe du langage C++, très utilisé par les informaticiens. Néanmoins, Java a été épuré des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, et l'héritage multiple remplacé par l'implémentation des interfaces. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.).

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM, JRE) applicatives multi plates-formes (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du code Java est assurée par la machine virtuelle. JRE –la machine virtuelle qui effectue la traduction et l'exécution du bytecode en code natif– supporte plusieurs processus de compilation (à la volée/bytecode, natif). La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.

Aujourd'hui, Java (sous la forme de JavaFx) trouve une nouvelle niche dans la création d'applications RIA (Rich Internet Applications), des applications qui proposent des fonctionnalités, notamment des interfaces, plus évoluées à la fois

sur Internet et sur les téléphones portables. Le langage JavaFx est un langage agile dérivé de Java, sous le contrôle d'Oracle, qui met à profit la portabilité de Java ainsi que les vastes bibliothèques déjà disponibles dans le langage Java.

## 1.2 Netbeans

NetBeans [29] est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)). L'IDE NetBeans s'appuie sur cette plateforme (voir figure 1).

## 2. Mesures de tests

L'estimation du processus de filtrage est une tâche plus que nécessaire pour la validation et la comparaison des systèmes de recommandations. Les algorithmes peuvent être évalués selon plusieurs critères, dans [Delgado, 99], [Massa, 09] l'auteur a classé ces métriques en deux catégories : les métriques *prédictives* et les métriques de *classification*.

Les métriques *prédictives* calculent la différence entre l'évaluation prédite par le système et l'évaluation réelle donnée par l'utilisateur, parmi les mesures les plus utilisées, on retrouve l'erreur moyenne absolue MAE (*Mean Absolute Error*), l'erreur moyenne quadratique MSE (*Mean Square Error*), la racine de l'erreur moyenne quadratique RMSE (*Root Mean Square Error*) et l'erreur moyenne de prédiction MPE (*Mean Prediction Error*).

Le MAE (5.1), vecteur d'erreur qui mesure la moyenne d'erreur entre les valeurs réelles portées par les utilisateurs et les valeurs prédites par le système vis-à-vis des items recommandés, sachant qu'il ya  $N$  utilisateurs donc, la valeur du MAE est donnée comme suit :

$$MAE = \frac{1}{N} \sum_{u=1}^N |p_{u,i} - r_{u,i}| \quad (7)$$

$p_{u,i}$  représente la valeur de prédiction de l'item  $i$  à l'utilisateur  $u$ ,  
 $r_{u,i}$  représente la valeur réelle de la notation de l'utilisateur  $u$  envers l'item  $i$ .

MSE, qu'élevé l'erreur au carré avant de faire la somme, cette mesure est utile lorsque qu'on veut donner plus de criticité aux erreurs importantes.

$$MSE = \frac{1}{N} \sum_{i=1}^N (p_{u,i} - r_{u,i})^2 \quad (8)$$

Dans la littérature, le *RMSE* (5.3), est largement utilisé, à la place du *MSE*, pour évaluer les systèmes de recommandation. Il est utilisé par le fameux concours *Netflix Prize* (<http://www.netflixprize.com/>) pour identifier les meilleurs algorithmes de filtrage (Bell et Koren, 2007b, 2007a, 2007c ; Bell, Koren et Volinsky, 2007c, 2007a, 2007b).

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (p_{u,i} - r_{u,i})^2} \quad (9)$$

Une autre variante, c'est le MAE normalisé (5.3) *NMAE* (*Normalized Mean Absolute Error*), cette mesure est équivalente à MAE, mais en prenant en considération l'échelle d'évaluation. Certains chercheurs préfèrent utiliser le *NMAE* plutôt que le MAE (Canny, 2002a ; Lemire, 2005). Son utilisation se justifie quand on doit procéder à des expérimentations utilisant différents corpus d'évaluation. En effet, cette mesure fournit une valeur normalisée du MAE facilitant la comparaison pour des échelles de notation différentes (Goldberg et al., 2001).

$$NMAE = \frac{1}{N} \sum_{i=1}^N \left| \frac{p_{u,i} - r_{u,i}}{E_{max} - E_{min}} \right| \quad (10)$$

Où  $E_{max}$  : la valeur max de l'échelle de notation ;

$E_{min}$  : la valeur min de l'échelle de notation.

Massa et Avesani [Massa, 09], ont introduit une autre mesure, *MAUE* (*Mean Absolute User Error*) sert à essimer l'erreur moyenne absolue pour chaque utilisateur puis calculer la moyenne pour l'ensemble des utilisateurs.

Par ailleurs, les métriques de classification appelées communément les métriques d'aide à la décision ont été héritées du domaine de recherche d'information. À l'opposé des mesures d'erreurs les métriques d'aide à la décision n'évaluent pas la qualité de la

prédiction, mais la pertinence des items [Adomavicius, 05], [Ziegler, 04]. Afin de mieux

interpréter ces mesures de classification, on prend pour point de départ le tableau répertoriant les classes des items.

**Tableau 3. Classes des items**

	Recommandés	Non recommandés	Total
Pertinents	$N_{pr}$	$N_{pnr}$	$N_p$
Non pertinents	$N_{npr}$	$N_{npnr}$	$N_{np}$
Total	$N_r$	$N_{nr}$	$N$

Où  $N_{pr}$ : représente le nombre des items pertinents recommandés.

$N_{pnr}$ : représente le nombre des items pertinents non recommandés.

$N_p$ : représente le nombre des items pertinents.

$N_{npr}$ : représente le nombre des items non pertinents recommandés.

$N_{npnr}$ : représente le nombre des items non pertinents non recommandés.

$N_{np}$ : représente le nombre des items non pertinents.

$N_r$ : représente le nombre des items recommandés.

$N_{nr}$ : représente le nombre des items non recommandés.

$N$ : représente le nombre des items.

A partir du tableau (3) on définit les mesures suivantes :

**Rappel** : représente un taux ou un nombre d'items pertinents retrouvés par rapport au nombre d'items pertinents que possède le système. Cela signifie que l'utilisateur souhaite voir apparaître tous les items intéressants (pertinents) pour lui. Si cette adéquation entre le nombre pertinent recommandé et le nombre pertinent est importante alors le taux de rappel est élevé. À l'inverse si le système possède de nombreux items intéressants mais que ceux-ci n'apparaissent pas dans la liste de recommandation, on parle de silence. Le silence s'oppose au rappel. (Miller, Konstan et Riedl, 2004 ; Faucher, 2007, 2008b).

$$R = \frac{N_{pr}}{N_p} \quad (11)$$

**Précision** : représente un taux ou un nombre d'items pertinents recommandés par rapport au nombre d'items recommandés. Les items superflus (non pertinents) retournés à l'utilisateur constituent le bruit qui s'oppose à la précision (Sarwar *et al.*, 2000b).

$$P = \frac{N_{pr}}{N_r} \quad (12)$$

**F-mesure** : est un compromis entre le rappel et la précision. Elle permet de comparer les performances des algorithmes par une seule mesure (Huang, Chen et Zeng, 2004 ; Montaner, 2003 ; Sarwar *et al.*, 2000b).

$$F = \frac{2.R.P}{R+P} \quad (13)$$

**Breese score** : une autre mesure proposée par Breese *et al.* (1998) comme une variante du taux de rappel fondée uniquement sur les items évalués, du fait que l'utilité d'une recommandation, se base essentiellement sur la probabilité que cette dernière soit consultée. Ainsi, cette mesure tient compte de l'ordre de présentation des recommandations en leur assignant différents poids (Stritt, Tso et Schmidt-Thieme, 2006; Ziegler *et al.*, 2005).

**Courbe ROC**: (*Receiver Operating Characteristic*): cette métrique mesure l'ordonnancement sous forme graphique avec le taux de vrais positifs en ordonnée et le taux de faux positifs en abscisse figure (5.1), (Good *et al.*, 1999 ; Melville, Mooney et Nagarajan, 2002; Rojsattarat et Soonthornphisaj, 2003 ).

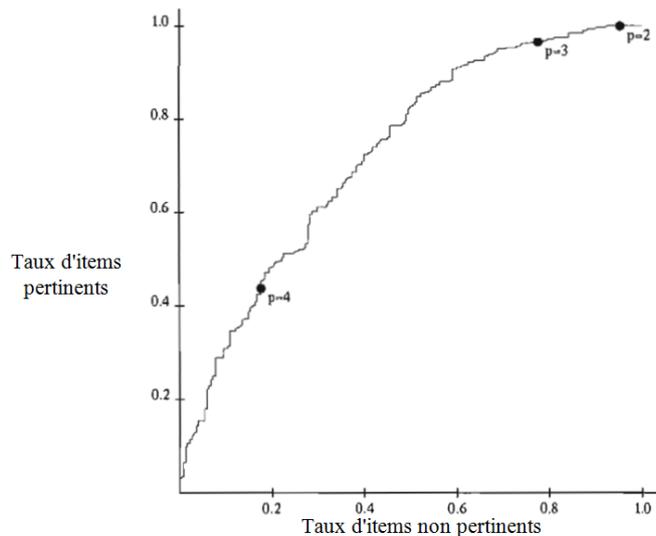


Figure 15. Graphe de ROC

**NDPM** (Normalized *Distance-based Performance Measure*): peu de recherche qu'utilise cette mesure qui montre le résultat ordonné comme la mesure ROC. Plus particulièrement, cette mesure estime la «distance» entre le rang de l'item évalué par l'utilisateur et le rang prédit par le l'algorithme de recommandation [Balabanovic, 97].

**NDCG** (*Normalised Discounted Cumulative Gain*) : est une mesure utilisée en recherche d'information qui utilise l'hypothèse que la pertinence diminue de façon logarithmique par rapport à la position dans la liste [Järvelin, 02].

### Autres métriques

Quoique les mesures d'erreurs et les métriques d'aide à la décision permettent d'évaluer les systèmes de filtrage, il existe ainsi d'autres aspects pratiquement négligés comme :

- **Couverture**: mesure le pourcentage des items pour lesquels le système a pu calculer des recommandations soit de façon globale ou à travers les items choisis par l'utilisateur.

- **Taux d'apprentissage** : les algorithmes d'apprentissage sont une composante importante des systèmes de filtrage. Leur efficacité varie en fonction de la quantité de données d'apprentissage mise en œuvre. Si la quantité de données d'apprentissage augmente, la qualité des recommandations s'augmente proportionnellement avec cette quantité. Or, cette métrique évalue la qualité des recommandations par rapport à la quantité de données d'apprentissage disponibles [Herlocker, 00].

- **Satisfaction des utilisateurs** : les recherches présentées par Cosley et al. [Rashid, 02] prouvent que la satisfaction de l'utilisateur est inversement proportionnelle vis-à-vis aux erreurs commises par le système de filtrage dans le processus de calcul de prédictions. Donc, le degré de satisfaction est un paramètre lié à la collaboration de l'ensemble et la performance de l'algorithme utilisé [Herlocker, 04].

- **Similarité moyenne** : cette métrique, introduite par Miller (2003), liée fortement à la qualité du voisinage. Plus précisément, la similarité moyenne est un facteur qui détermine la similarité moyenne du groupe d'utilisateurs choisis pour le calcul de la recommandation finale [Herlocker, 04] Pour notre étude expérimentale, nous avons retenus les mesures *RMSE*, *MAE*, *rappel* et *précision* pour évaluer et comparer les algorithmes implémentés.

### 3. Jeux de Données

#### 3.1 MovieLens

MovieLens est un système de recommandation basé sur le Web et une communauté virtuelle qui recommande aux utilisateurs de regarder des films, en fonction de leurs préférences en matière de film, en utilisant un filtrage collaboratif des classements et

des critiques de films par les membres. Il contient environ 11 millions de votes pour environ 8500 films. MovieLens a été créé en 1997 par GroupLens Research, un laboratoire de recherche du département d'informatique et d'ingénierie de l'université du Minnesota [SCH03], afin de recueillir des données de recherche sur les recommandations personnalisées [JUL97].

MovieLens base ses recommandations sur les informations fournies par les utilisateurs du site Web, telles que les classifications de films. [SCH03] Le site utilise divers algorithmes de recommandation, y compris des algorithmes de filtrage collaboratif tels que l'approche item-item, [SAR01] utilisateur-utilisateur. De plus, pour résoudre le problème du démarrage à froid des nouveaux utilisateurs, MovieLens demande aux nouveaux utilisateurs d'indiquer à quel point ils aiment regarder différents groupes de films (par exemple, des films avec humour noir ou des comédies romantiques). Les préférences enregistrées par cette enquête permettent au système de formuler des recommandations initiales, même avant que l'utilisateur n'ait évalué un grand nombre de films sur le site Web.

GroupLens Research, un laboratoire de recherche sur les interactions homme-ordinateur de l'Université du Minnesota, fournit les ensembles de données d'évaluation rassemblés sur le site Web de MovieLens à des fins de recherche. L'ensemble de

données complet contient 26 000 000 évaluations et 750 000 applications de balises appliquées à 45 000 films par 270 000 utilisateurs. Il comprend également des données sur le génome des étiquettes avec 12 millions de scores de pertinence sur 1 100 étiquettes (dernière mise à jour le 8/2017). De nombreux types de recherche sont basés sur les ensembles de données MovieLens.

On va travailler avec des données issues du système de recommandation de films MovieLens. Plusieurs jeux de données sont disponibles sur le site de GroupLens. Ces jeux de données diffèrent par leur taille et nous travaillerons avec le plus petit.

MovieLens est un site communautaire de recommandation de films. Les utilisateurs de ce site notent des films de 1 à 5. Ils peuvent également demander des suggestions de films étant donné les notations qu'ils ont fournies.

Chaque utilisateur est décrit par quelques attributs :

- un numéro d'identification,
- son âge,
- son sexe,
- son métier,
- son code postal.

Chaque film est décrit par :

- un numéro d'identification,
- son titre,
- sa date de sortie au cinéma
- son genre.

On dispose alors d'une liste de notes sous la forme :

- identifiant d'utilisateur
- identifiant de film
- note attribué par cet utilisateur à ce film,
- information de date : nombre de secondes écoulées depuis le 1/1/1970.

Il y a 100.000 notes, provenant de 943 utilisateurs pour 1682 films. Les autres jeux de données disponibles sur ce site web en comptent respectivement 1 million et 10 millions.

L'objectif est de pouvoir proposer des recommandations à un utilisateur et mesurer la précision de système de recommandation.

## 4. Résultats et discussion

Dans cet article, la similarité entre les utilisateurs a été calculée sur trois matrices d'évaluations de (10 x 10), (20 x 20) et un dataset MovieLens (900 x 1683) en utilisant certaines des méthodes de mesure de la similarité mentionnées dans le chapitre 2.

Les préférences utilisateur pour les éléments peuvent être représentées comme une matrice utilisateur-item  $R_{N \times M}$  où  $N$  est le nombre d'utilisateurs et  $M$  le nombre d'items.  $U$  représente l'ensemble de tous les utilisateurs et  $P$  représente l'ensemble de tous les items.

### 4.1 Déroulement de l'exécution

L'Exécution se fait en quatre étapes :

1. Choix de la technique à utiliser (Collaborative ou Prédiction des liens).
2. Choix du jeu de données et de l'algorithme (Pearson, Cosine, Jaccard...etc).
3. Calcule des Prédictions.
4. Calcul du MAE (Evaluation).

### 4.2 Présentation des interfaces



Figure 16. Fenêtre principale

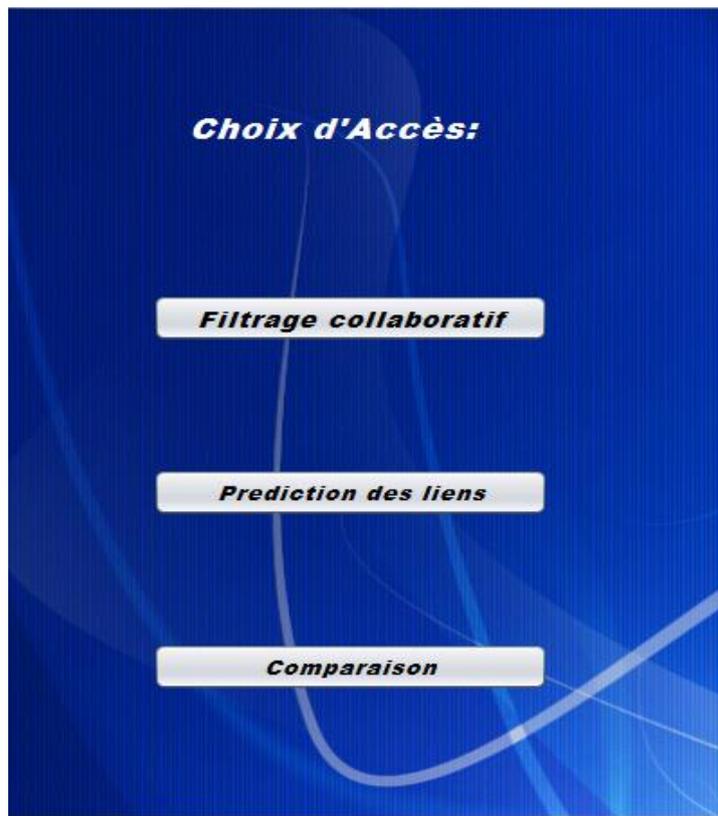


Figure 17. Fenêtre d'accès

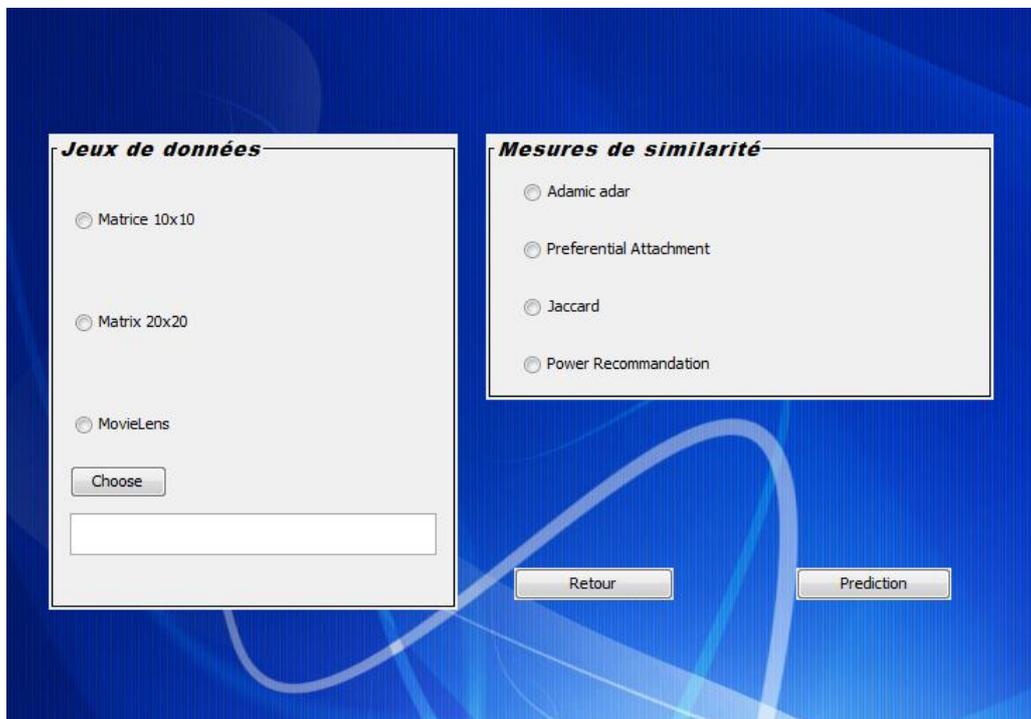
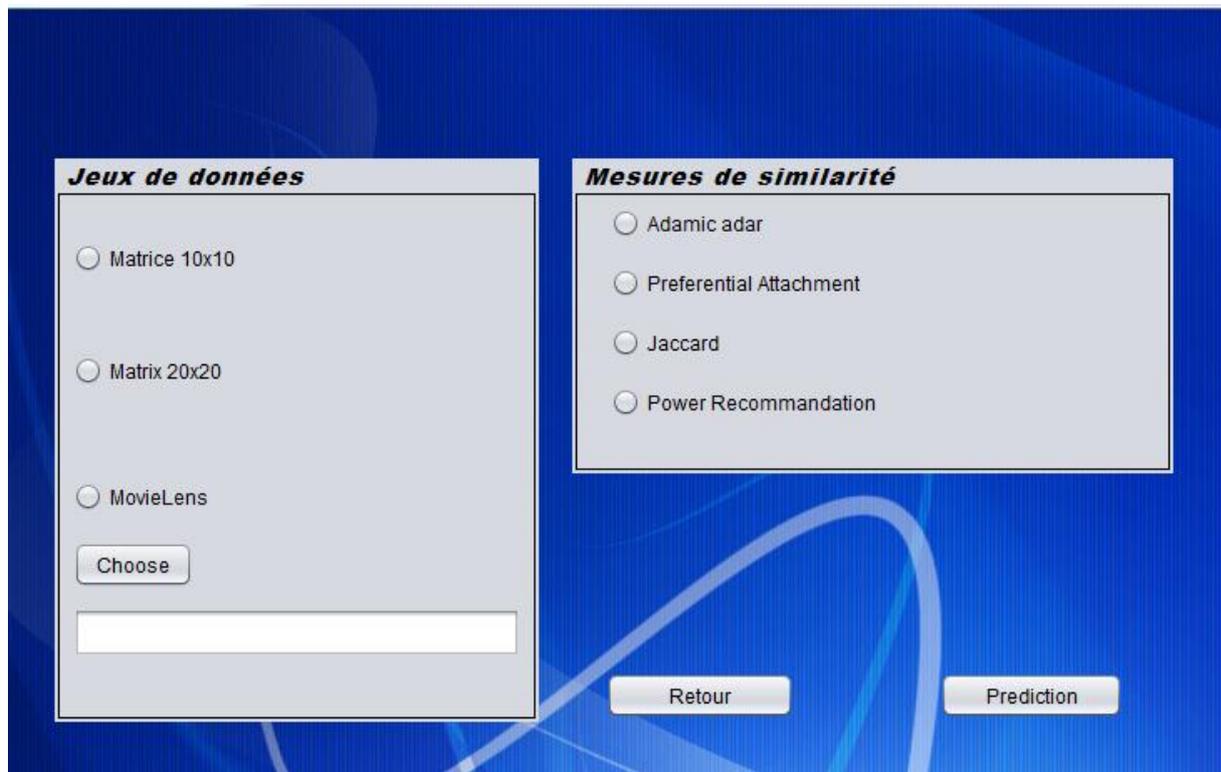


Figure 18. Fenêtre filtrage collaboratif



**Figure 19. Fenêtre prediction des liens**

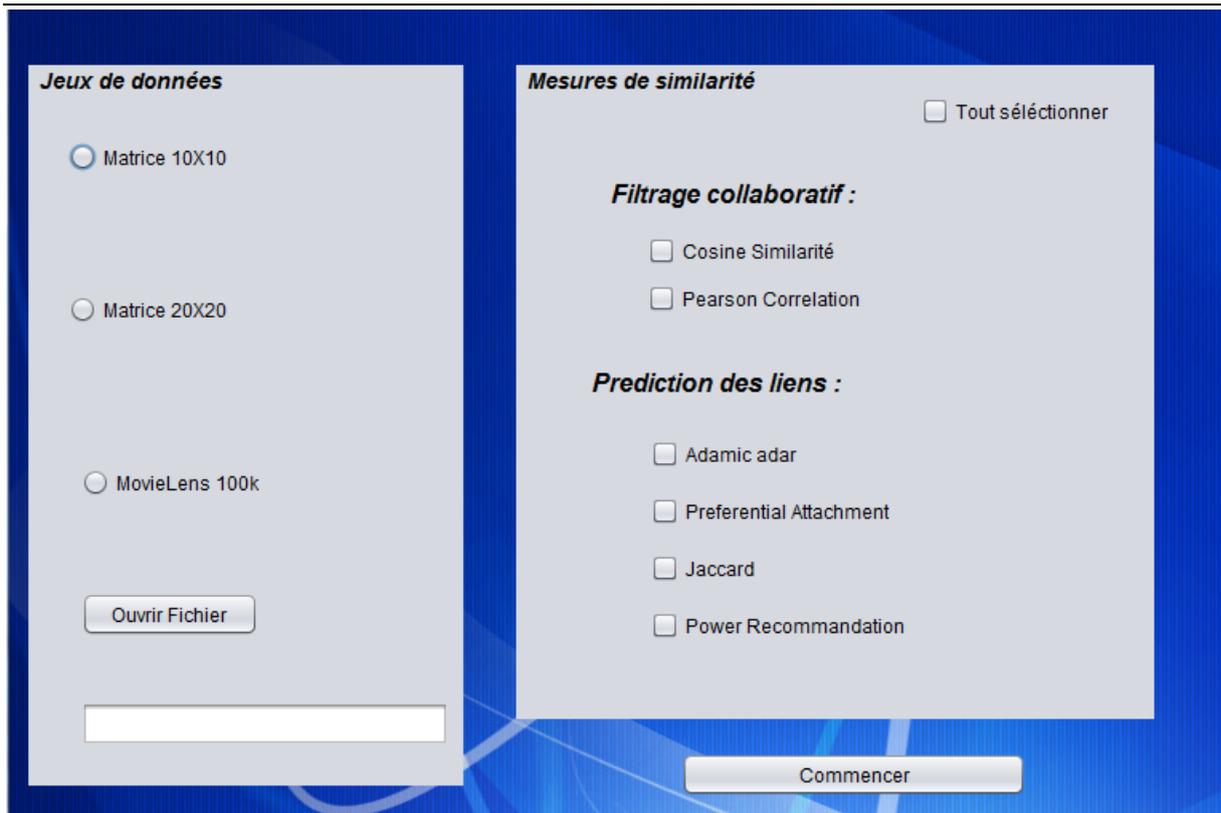


Figure 20. Fenêtre comparaison graphique

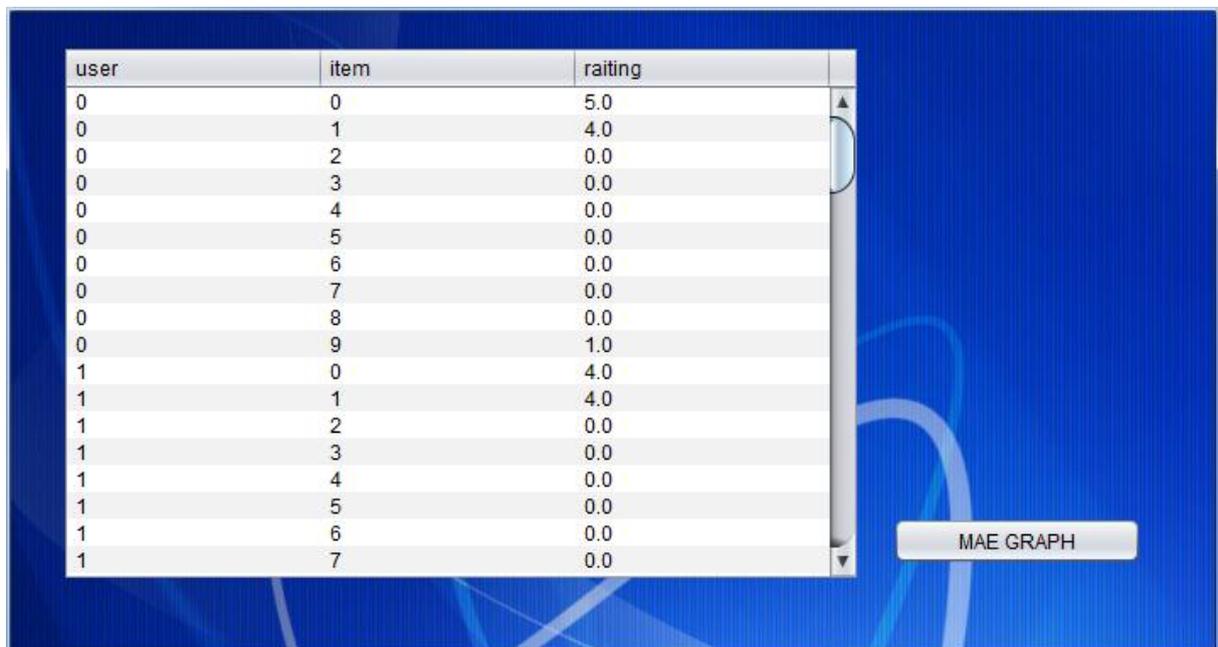


Figure 21. Fenêtre résultats des prédictions

## Résultats du MAE

### a. Matrice 10 x 10

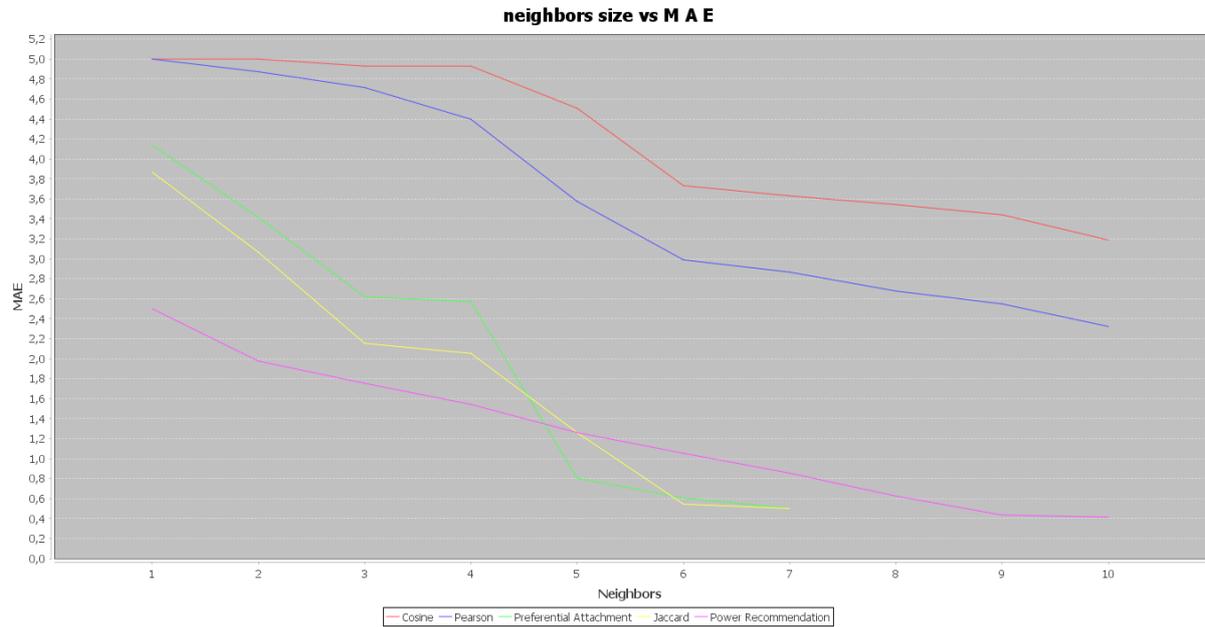


Figure 22. Résultats du MAE (Matrice 10x10)

### Matrice 20 x 20

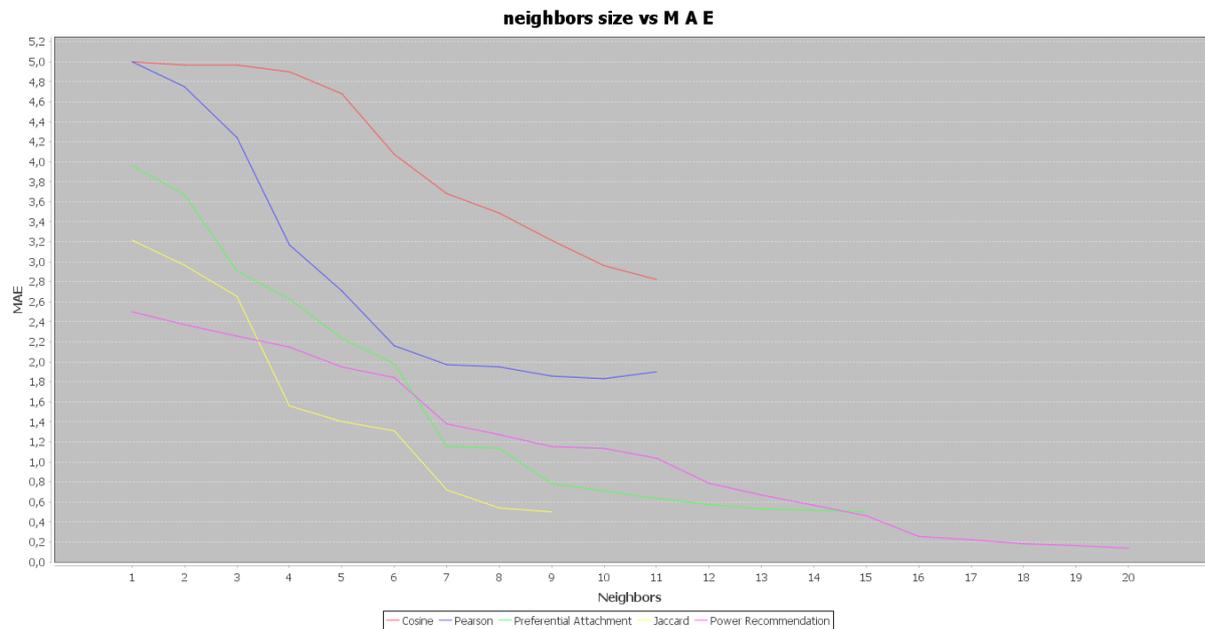


Figure 23. Résultats du MAE (Matrice 20x20)

## Conclusion Générale

Dans ce travail, nous nous intéressons aux systèmes de filtrage collaboratif qui sont sans doute le type le plus utilisé, vu les avantages qu'ils procurent, comme la possibilité de recommander tout type de ressources (images, vidéos etc.), la possibilité d'exprimer d'autres facteurs liés à la qualité, au public visé etc. On constate néanmoins certains inconvénients de cette technique, incluant le démarrage à froid, la masse critique, le rapport coût-bénéfice et l'expression limitée du besoin.

Les méthodes de filtrage collaboratif initiales étaient statistiques, bien que plusieurs chercheurs aient formulé le filtrage collaboratif comme un problème d'apprentissage.

Le filtrage collaboratif doit aujourd'hui tenir compte de données incomplètes, voire manquantes, concernant les votes sur les items, C'est pourquoi les méthodes purement statistiques sont parfois jugées inadaptées. Deux classes majeures d'algorithmes sont identifiées pour résoudre ce problème : les algorithmes basés mémoire et ceux basés modèle.

Les techniques basées sur la mémoire présentent l'avantage d'être très réactives, en intégrant immédiatement au système les modifications des profils utilisateurs, cependant, la complexité des algorithmes en temps et en mémoire est beaucoup trop importante pour les grosses bases de données. Les algorithmes basés modèle minimisent le problème de la complexité algorithmique en mémoire. Par ailleurs, ils perçoivent dans ces modèles une valeur ajoutée au-delà de la seule fonction de prédiction. Cependant, ces méthodes ne sont pas assez dynamiques et elles réagissent mal à l'insertion de nouveaux contenus dans la base de données. De plus, elle nécessite une phase d'apprentissage à la fois pénalisante pour l'utilisateur et coûteuse en temps de calcul pour les grosses bases de données.

À travers ce travail, nous avons essayé d'atténuer le problème de filtrage collaboratif qui est la matrice creuse à travers l'utilisation des algorithmes de prédiction des liens dans les graphes.

## Bibliographie

[ADO11] : Adomavicius, G. and Tuzhilin, A. (2011).Context-aware recommender systems. Recommender systems handbook, pages 217–253. Springer.

[AMA11]: D'Amato A, et al. (2011) Mehercules, adhuc Bacchus! The debate on wine proteomics continues.

[BEN17] : Idir Benouaret. Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels. Université de Technologie de Compiègne, 2017.

[BER03] : C. BERRUT , « Filtrage collaboratif » , Assistance intelligente à la recherche d'informations , Chapitre 8 , Hermes science , 2003.

[BIS03] : C. BISIAUX , « Expérience sur l'utilisation de conjonctions de termes et la prise en compte des dépendances entre termes d'indexation dans le cadre du filtrage adaptatif » , Thèse dirigée par : C. BROUARD , MRIM , Juin 2003.

[DAH14] : Mohamed Ryadh DAHIMENE, Filtrage et Recommandation sur les Réseaux Sociaux, 2014, Ecole Doctorale Informatique, Télécommunication et Electronique (paris).

[DAO07] : M. Daoud, Recherche contextuelle d'information. RJCRI 2007.

[DAN99] : Daniel Billsus and Michael J. Pazzani. A hybrid user model for news story classification. In UM '99 : Proceedings of the seventh international conference on User modeling, pages 99 108, Secaucus, NJ, USA, 1999. Springer-Verlag New York, Inc.

[DEN04] : N. DENOS, C. BERRUT, L. GALLARDO-LOPEZ , A. NGUYEN , « COCoFil : une plateforme de filtrage collaboratif orientée vers la communauté » , Laboratoire CLIPS-IMAG , CORIA 2004.

[GAL05] : M. L. GALLARDO LOPEZ, « Accès à l'Information par un Système de Filtrage Collaboratif Contrôlé », Université Joseph Fourier - Grenoble I, Laboratoire CLIPS – IMAG, janvier 2005.

[GIL03] : Bas van Gils and Eric D. Schabell. User-pro les for information retrieval. In BNAIC 03 : Proceedings of the 15th Belgian-Dutch Conference on Artificial Intelligence, 2003.

[GOL03] : Goldberg, D.S. and Roth, F.P. Assessing experimentally derived interactions in a small world. In Proceedings of the National Academy of Sciences USA, 2003.

[HER04]: M. R. McLaughlin, J. L. Herlocker.A Collaborative Filtering Algorithm andEvaluation Metric that Accurately Model the User Experience. SIGIR

'04: Pro-ceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press, (2004).

[HUN05] : Miha Grcar, Hunja Mladenic, and Marko Grobelnik. User profiling for interest- focused browsing history. In Proceedings of Workshop on User Aspects of the Semantic Web, 2005.

[HUA04] : Huang, Z., Chen, H. and Zeng, D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering, 2004.

[JUL97] : Ojeda-Zapata, Julio (1997-09-15). "New Site Personalizes Movie Reviews". St. Paul Pioneer Press. p. 3E.

[KAR10] : Karatzoglou, A., Amatriain, X., Baltrunas, L., and Oliver, N. (2010). Multiverse recommendation : n-dimensional tensor factorization for context-aware collaborative filtering. In Proceedings of the fourth ACM conference on Recommender systems, pages 79–86. ACM.

[KAZ04] : Kazunari Sugiyama, Kenji Hatano, and Masatoshi Yoshikawa. Adaptive web search based on user profile constructed without any effort from users. In WWW '04 : Proceedings of the 13th international conference on World Wide Web, pages 675–684, New York, NY, USA, 2004. ACM Press.

[LEE05] : J. S. LEE, C.H. JUN, J. LEE, S. KIM, « Classification-based collaborative filtering using marketbasket data », Department of Industrial Engineering, Pohang University of Science and Technology, South Korea 2005.

[LEV14]: Claudio Adrian Levinas, An Analysis of Memory Based Collaborative Filtering Recommender Systems With Improvement Proposals, 2014, UPC.

[LET95] : H. Lieberman. Letizia : An agent that assists web browsing. In Proceedings of the International Joint Conference on Artificial Intelligence, Montreal, August 1995.

[LIR98] : Liren Chen and Katia Sycara. Webmate : a personal agent for browsing and searching. In AGENTS '98 : Proceedings of the second international conference on Autonomous agents, pages 132–139, New York, NY, USA, 1998. ACM Press.

[LOM09] : S. Lombardi, S. S. Anand, M. Gorgoglione. Context and Customer Behavior in Recommendation. Septembre 2009.

[MIR04] : Mirco Speretta. Personalizing search based on user search histories. Master's thesis, University of Kansas, 2004.

[MIQ03] : Miquel Montaner, Beatriz López, and Josep Lluís De La Rosa. A taxonomy of recommender agents on the internet. Artif. Intell. Rev., 19(4) :285–330, 2003.

[MAL95] : D.A. MALTZ , K. EHRLICH , « Pointing the way : active collaborative filtering », Proceedings of CHI'95, Mai 1995.

[PHI03] : Hyoung R. Kim and Philip K. Chan. Learning implicit user interest hierarchy for context in personalization. In IUI '03 : Proceedings of the 8th international conference on Intelligent user interfaces, pages 101 108, New York, NY, USA, 2003. ACM Press.

[PAZ96] : M. Pazzani, J. Muramatsu, and D. Billsus. Syskill & Webert : Identifying interesting web sites. In Proc. AAAI Spring Symposium on Machine Learning in Information Access, 1996.

[ROC71] : J. Rocchio. Relevance Feedback in Information Retrieval. G. Salton (editor), The SMART Retrieval System : Experiments in Automatic Document Processing. Prentice Hall, Inc., Englewood Cliffs, NJ, 1971.

[SOL12] : Soltani Réda, Impact de la prise en compte d'information contextuelle sur la pertinence des systèmes de recommandations, 2012, Université d'Oran.

[SCH03]: Schofield, Jack (2003-05-22). "Land of Gnod". The Guardian. London.

[SAR01] : Sarwar, Badrul, et al. "Item-based collaborative filtering recommendation algorithms." Proceedings of the 10th international conference on World Wide Web. ACM, 2001.

[TMA01] : TMAR, M. BOUGHANEM , « Filtrage d'information par combinaison d'un profil positif et profil négatif » , IRI/SIG , Compus Univ Toulouse III , Université de Nantes-Paris X , 3° congré du chapitre français de l'ISKO , juillet 2001.

[THA08] : Thanh Trung Van. Utilisation de profils utilisateurs pour l'accès à une bibliothèque numérique. Bibliothèque électronique [cs.DL]. Ecole Nationale Supérieure des Mines de Saint-Etienne, 2008. Français.

[THO05] : Thorsten Joachims, Laura Granka, Bing Pan, Helene Hembrooke, and Geri Gay. Accurately interpreting clickthrough data as implicit feedback, New York, NY, USA, 2005. ACM Press.

[ZHA00] : Young-Woo Seo and Byoung-Tak Zhang. A reinforcement-learning agent for personalized information filtering. In IUI '00 : Proceedings of the 5th international conference on Intelligent user interfaces, pages 248 251, New York, NY, USA, 2000. ACM Press.

[ZHO07] Zhou,T., Bipartite network projection and personal recommendation. Physical Review ,2007.